PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-181676
(43)Date of publication of application : 23.07.1993

(51)Int.Cl. G06F 9/38
G06F 9/38
G06F 9/38
G06F 11/00
G06F 15/16

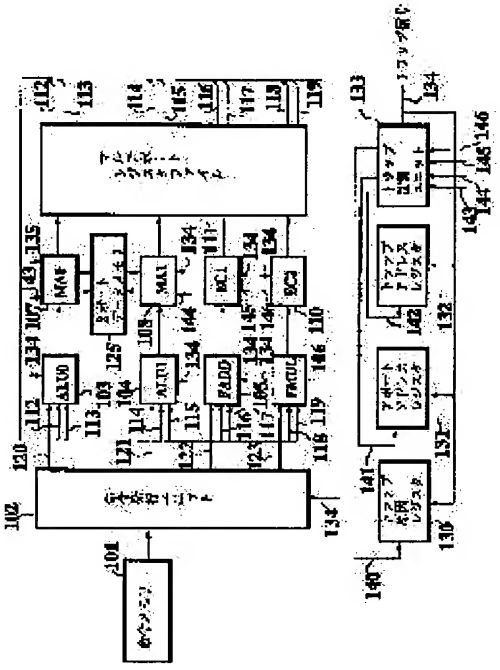
(21)Application number : 04-082490
(22)Date of filing : 03.04.1992

(71)Applicant : TOSHIBA CORP
(72)Inventor : AIKAWA TAKESHI
MINAGAWA KENJI
SAITO MITSUO
TAKEDA JOJI

(30)Priority
Priority number : 03 73364 Priority date : 05.04.1991 Priority country : JP

(54) PARALLEL PROCESSING TYPE PROCESSOR SYSTEM

(57)Abstract:
PURPOSE: To provide the parallel processing type processor system such as a super scolar processor, etc., integrating a trap, which can be processed without prolonging cycle time, and a stall control function.
CONSTITUTION: This system is provided with N (N is an integer) computing elements 103-106 to execute plural instructions at the same time, instruction supplying means 102 for supplying the instructions to be executed by the N computing elements 103-106, and trap control means for controlling the N computing elements 103-106 so as to abort all the processing of M (M≥N, M is an integer) instructions simultaneously supplied to the N computing elements 103-106 when a processing exception is generated in the case of executing any one of these M instructions while supplying the M instructions from the instruction supplying means 102 to the N computing elements 103106 at the same time.
Thus, the control of the trap and stall can be more efficiently executed without prolonging the cycle time and lowering the clock frequency of the system.



(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号

特開平5－181676

(43)公開日　平成5年(1993)7月23日

(51)Int.Cl.[5]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	9/38	３８０　Ｂ	9290－5B		
		３１０　Ｘ	9290－5B		
		３７０　Ｘ	9290－5B		
	11/00	３１０　Ｇ	7313－5B		
	15/16	３９０　Ｚ	9190－5L		

審査請求　未請求　請求項の数２(全 26 頁)

(21)出願番号　特願平4－82490

(22)出願日　平成４年(1992)４月３日

(31)優先権主張番号　特願平3－73364

(32)優先日　平３(1991)４月５日

(33)優先権主張国　日本（ＪＰ）

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者　相川　健
神奈川県川崎市幸区小向東芝町１　株式会社東芝総合研究所内

(72)発明者　皆川　健二
神奈川県川崎市幸区小向東芝町１　株式会社東芝総合研究所内

(72)発明者　斎藤　光男
神奈川県川崎市幸区小向東芝町１　株式会社東芝総合研究所内

(74)代理人　弁理士　三好　秀和　(外１名)

最終頁に続く

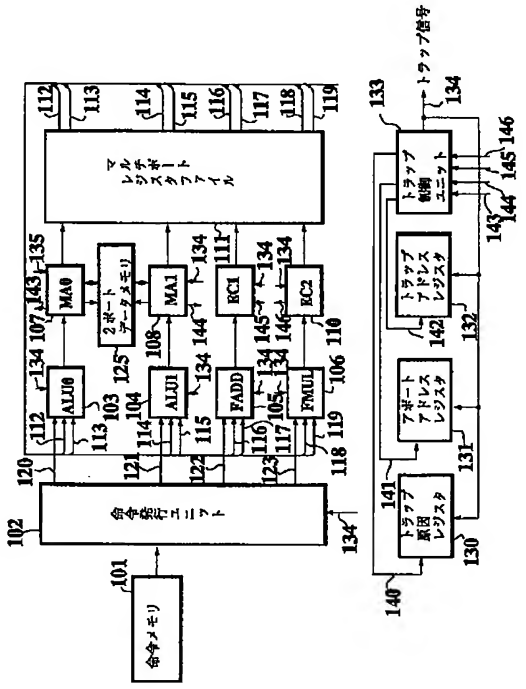
(54)【発明の名称】　並列処理型プロセッサシステム

(57)【要約】

【目的】サイクル時間を増加することなく処理可能なトラップとストールの制御機能を組み込んだスーパースカラプロセッサ等の並列処理型プロセッサシステムを提供すること。

【構成】複数命令を同時実行するＮ個（Ｎは整数）の演算器と、前記Ｎ個の演算器により実行される命令を供給する命令供給手段と、Ｍ個（Ｎ≧Ｍ、Ｍは整数）の命令が前記命令供給手段から前記Ｎ個の演算器に同時に供給され、これらＭ個の命令中の少なくとも一つの命令実行において処理例外が発生したとき、前記Ｎ個の演算器に同時に供給された前記Ｍ個の命令の処理を全てアボートするように前記Ｎ個の演算器を制御するトラップ制御手段と、を備える並列処理型プロセッサシステム。

【効果】サイクル時間を増加させることなく、スーパースカラプロセッサ等の並列処理型プロセッサシステムにおいてシステムのクロック周波数を低下させずにトラップとストールの制御をより効率的に行うことが可能となる。



1

【特許請求の範囲】

【請求項１】　複数命令を同時実行するＮ個（Ｎは整数）の演算器と、

前記Ｎ個の演算器により実行される命令を供給する命令供給手段と、

Ｍ個（Ｎ≧Ｍ、Ｍは整数）の命令が前記命令供給手段から前記Ｎ個の演算器に同時に供給され、これらＭ個の命令中の少なくとも一つの命令実行において処理例外が発生したとき、前記Ｎ個の演算器に同時に供給された前記Ｍ個の命令の処理を全てアボートするように前記Ｎ個の演算器を制御するトラップ制御手段と、

を備えたことを特徴とする並列処理型プロセッサシステム。

【請求項２】　Ｎ個の演算器にＭ個（Ｎ≧Ｍ）の命令を同時に供給するステップと、

これらＭ個の命令中の少なくとも一つの命令実行において処理例外が発生したとき、前記Ｎ個の演算器に同時に供給された前記Ｍ個の命令の処理を全てアボートするように前記Ｎ個の演算器を制御するステップと、

を備えたことを特徴とする並列処理型プロセッサシステムの制御方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、命令レベルでの並列実行を行うための並列処理型プロセッサシステムに関し、より詳細には、並列処理型プロセッサシステムにおけるトラップ処理とストール処理の制御機能に関するものである。

【０００２】

【従来の技術】近年、高速パーソナルコンピュータに対して高まりつつある要望に応るものとして、機械語命令レベルで並列処理が可能なスーパースカラプロセッサあるいはＶＬＩＷと呼ばれるＣＰＵが開発されＶＬＳＩチップとして実現されている。この様な並列処理ＣＰＵにおいては、ＲＩＳＣの命令を基本的な命令セットとして用い、複数命令を同時にフェッチ、実行することにより、処理性能を向上させている。特に、スーパースカラプロセッサは、従来の命令レベルで逐次処理を行うＲＩＳＣを実現しユーザプログラムレベルで互換性を保つことが可能なようなアーキテクチャを有し、計算機ユーザーからの期待が高い。

【０００３】このような従来のトラップ制御方法を採用する命令レベルで並列処理を行うことができるプロセッサシステムは、図１９に示すような構成を有している。

【０００４】この図１９の構成は、命令レベルで並列処理を実行可能なプロセッサシステムを実現するものであり、Ｆステージ（フェッチ）、Ｄステージ（デコード）、Ｅステージ（実行）、Ｍステージ（メモリアクセス）及びＷステージ（レジスタライトバック）の５段のパイプラインステージを有し、各命令の長さが１ワード

2

長（３２ビット）である。

【０００５】図１９に示されるように、このプロセッサシステムは、命令を格納するための命令メモリ１と；Ｆステージで４ワードバウンダリの４つの命令を命令メモリ１から同時にフェッチし、Ｄステージで４つのフェッチされた命令間のデータ依存関係及び制御依存関係を考慮し、Ｅステージで命令供給線２０、２１、２２及び２３を介して実行可能な命令を供給する命令発行ユニット２と；命令供給線２０及び２１から供給される命令に従って、Ｅステージで算術論理演算及びメモリアドレス計算を実行する算術論理演算ユニット（ALUO及びALU1）３及び４と；命令供給線２２から供給される命令に従ってＥステージで浮動小数点加減算を行う浮動小数点加算器（FADD）５と；命令供給線２３から供給される命令に従ってＥステージで浮動小数点乗除算を行う浮動小数点乗算器（FMUL）６と；ALUO３及びALU1４の出力に従って、Ｍステージで２ポートデータメモリ２５に対してメモリアクセス処理を行うメモリアクセスユニット（MAO 及びMA1 ）７及び８と；FADD５及びFMUL６の出力に従って、各々のＭステージでの浮動小数点計算における例外チェックを行うための浮動小数点例外チェックユニット（EC1 及びEC2 ）９及び１０と；ＷステージでMAO ７、MA1 ８、EC1 ９及びEC2 １０の出力を受ける４つの書き込みポートと、Ｅステージでオペランドデータ供給線１２～１９を介してALUO３、ALU1４、FADD５及びFMUL６へオペランドデータを供給するための８つの読み込みポートとからなる１２のポートを有するマルチポートレジスタファイル１１と、を備えている。

【０００６】この図１９の構成においては、MAO ７及びMA1 ８によってページフォールトやオーバーフローなどの算術演算例外トラップが発生し、又、EC1 ９及びEC2 １０によって浮動小数点演算例外トラップが発生する。

【０００７】このような例外トラップに対処するために、このプロセッサシステムには更に、トラップ発生の原因を格納するためのトラップ原因レジスタ３０と、トラップ発生を引き起こした命令のアドレスを格納するためのトラップアドレスレジスタ３２と、トラップ要求信号線４３～４６を介して送られる、MAO ７、MA1 ８、EC1 ９及びEC2 １０からのトラップ原因を受け、それに応じてトラップ信号をトラップ信号線３４－３８にアサートし、信号線４０及び４２を介してトラップ原因レジスタ３０及びトラップアドレスレジスタ３２への入力を適宜発生させる、トラップ制御ユニット３３と、を備えている。

【０００８】トラップ信号線３４におけるトラップ信号は、命令発行ユニット２、ALUO３及びALU1４へ送られ、一方、トラップ信号線３５～３８におけるトラップ信号は、各々、MAO ７、MA1 ８、EC1 ９及びEC2 １０へ送られる。トラップ制御ユニット３３からのトラップ信号に応じて、実行無効化フラグが各部に立てられ、その後の

パイプラインステージでの命令の処理をアボートする一方、命令発行ユニット２は予め定められたトラップ処理ルーチンに関する命令フェッチを開始し、このトラップ処理ルーチンにおいて、トラップ原因レジスタ３０及びトラップアドレスレジスタ３２に格納されたトラップ原因及びトラップアドレスが使用される。

【０００９】更に詳細には、トラップ制御ユニット３３は図２０に示されるような構成を有している。即ち、トラップ制御ユニット３３は更に、Mステージで現在実行される命令のアドレスの下位２ビットを除くことによって得られるワードアドレスの共通部分を格納するMステージプログラムカウンタ（MPC ）５１と；MステージでMA0 7、MA1 8、EC1 9及びEC2 １０によって現在実行される命令のアドレスの下位２ビットによって示されるワードアドレスの個別部分を格納するためのMステージサブプログラムカウンタ（submpc1，submpc2，submpc3及びsubmpc4 ）５３、５４、５５及び５６と；Mステージサブプログラムカウンタ５３～５６の中で最小のエントリを、MPC ５１のエントリと組み合わされてトラップアドレスレジスタ３２に供給されるトラップアドレス４２を発生するための出力４７として出力し、最小のエントリを有するMステージサブプログラムカウンタ５３～５６に対応するトラップ要求信号線４３～４６のうちの１つを介して送られるトラップ原因をトラップ原因レジスタ３０に供給されるトラップ原因４０として出力するトラップデータ生成ユニット５７を備えている。

【００１０】ここで、トラップ信号線３４におけるトラップ信号はトラップ要求信号線４３～４６のいずれか１つからトラップ原因を受けた時にアサートされ、各トラップ信号３５～３８は、トラップ要求信号線４３～４６の１つからトラップ要求を受け、対応するMステージサブプログラムカウンタ５３～５６の１つが、トラップ要求を発生したMステージサブプログラムカウンタ３５～５６のうちの１つにおけるエントリより大きいか又は等しいエントリを有する時にアサートされる。

【００１１】図２１は図１９のプロセッサシステムによって実行されるプログラムの一例を示す図であり、図２２は、上述の従来のトラップ制御方法を用いた図１９のプロセッサシステムによるパイプライン処理において、図２１のプログラムが実行された時に「load」命令でページフォールトが起こった場合の進行状況を示すものであり、斜線部分はアボートされた命令を示す。図２２に示すように、従来のトラップ制御においては、プログラム実行の進行中にn+2 番目の「load」命令の実行によりトラップが発生したとき、命令番号がn+2 より大きいか又は等しい命令だけがアボートされる。

【００１２】

【発明が解決しようとする課題】しかしながら、このような従来のトラップ制御方法においては、C+3 サイクル目にがMA1 8によってページフォールトが検出されてトラップ要求がトラップ要求線４４を介して示されたときに、トラップ信号線３５～３８におけるトラップ信号はMステージサブプログラムカウンタ５３～５６におけるエントリが互いに比較されてどれがどれより大きいかが決定されるまで決められず、そのような比較処理を施すサイクル時間がかなり長くなるためにクロック周波数の低下を引き起こすという問題があった。

【００１３】又、ＲＩＳＣは、単純なデータパス及び簡易な制御回路を有する構成を必要としており、複数のＲＩＳＣデータパスを有するスーパースカラプロセッサのデータパスはさほど複雑ではないが、スーパースカラプロセッサの制御回路は、命令供給制御等を必要とするために非常に複雑となる。特に、ＯＳのようなソフトウエアのサポートがないと処理の続行が不可能になる、いわゆる例外と呼ばれるケースの処理のためのハードウェアが非常に複雑になり、そのようなハードウエアの設計に非常に時間を要するために、このようなハードウェアがスーパースカラプロセッサの実現におけるクリチカルパスとなる場合が多いという問題があった。

【００１４】本発明は、この様な従来技術の課題を解決するためになされたもので、サイクル時間を増加させることなく処理可能で、システムにおけるクロック周波数の低下を防げるようなトラップ及びストール制御機能を組み込んだ、スーパースカラプロセッサ等の並列処理型プロセッサシステムを提供することを目的とするものである。

【００１５】

［発明の構成］

【００１６】

【課題を解決するための手段】上記課題を解決するため本発明の並列処理型プロセッサシステムは、複数命令を同時実行するＮ個（Ｎは整数）の演算器と、前記Ｎ個の演算器により実行される命令を供給する命令供給手段と、Ｍ個（Ｎ≧Ｍ、Ｍは整数）の命令が前記命令供給手段から前記Ｎ個の演算器に同時に供給され、これらＭ個の命令中の少なくとも一つの命令実行において処理例外が発生したとき、前記Ｎ個の演算器に同時に供給された前記Ｍ個の命令の処理を全てアボートするように前記Ｎ個の演算器を制御するトラップ制御手段とを備えたことを特徴とする。

【００１７】又、本発明の並列処理型プロセッサシステムの制御方法はＮ個の演算器にＭ個（Ｎ≧Ｍ）の命令を同時に供給するステップと、これらＭ個の命令中の少なくとも一つの命令実行において処理例外が発生したとき、前記Ｎ個の演算器に同時に供給された前記Ｍ個の命令の処理を全てアボートするように前記Ｎ個の演算器を制御するステップとを備えたことを特徴とする。

【００１８】

【作用】本発明においては、命令供給手段によりＮ個の演算器に同時に供給されたＭ個の命令の中の１つ以上の

命令実行過程で処理例外が発生したとき、同時に供給されたＭ個の命令の実行をすべて中止する手段を備えたので、同時に演算器に供給されたＭ個の命令の処理過程でどれか１つでも例外を発生したらＭ個の命令をすべてアボートしてＯＳ内の処理ルーチンへディスパッチすることにより、例外処理のためのハードウエアが簡単化される。

【００１９】

【実施例】以下、図１から図３を参照して、本発明に係る並列処理型プロセッサシステムの第１の実施例を詳細に説明する。

【００２０】この図１の構成は、命令レベルで並列処理を行うことができるプロセッサシステムを実現するものであり、Ｆステージ（フェッチ）、Ｄステージ（デコード）、Ｅステージ（実行）、Ｍステージ（メモリアクセス）及びＷステージ（レジスタライトバック）の５段のパイプラインステージがあり、各命令の長さは１ワード長（３２ビット）である。

【００２１】この第１実施例においては、図１に示すように、プロセッサシステムは、命令を格納するための命令メモリ１０１と；Ｆステージで４ワードバウンダリの４つの命令を命令メモリ１０１から同時にフェッチし、Ｄステージで４つのフェッチされた命令間のデータ依存関係及び制御依存関係を考慮し、Ｅステージで命令供給線１２０、１２１、１２２及び１２３を介して実行可能な命令を供給するためのする命令発行ユニット１０２と；命令供給線１２０及び１２１から供給される命令に従って、Ｅステージで算術論理演算及びメモリアドレス計算を実行する算術論理演算ユニット（ALU0及びALU1）１０３及び１０４と；命令供給線１２２から供給される命令に従ってＥステージで浮動小数点加減算を行う浮動小数点加算器（FADD）１０５と；命令供給線１２３から供給される命令に従ってＥステージで浮動小数点乗除算を行う浮動小数点乗算器（FMUL）１０６と；ALU0１０３及びALU1１０４の出力に従って、Ｍステージで２ポートデータメモリ１２５に対するメモリアクセス処理を行うメモリアクセスユニット（MA0 及びMA1 ）１０７及び１０８と；FADD１０５及びFMUL１０６の出力に従って、各々のＭステージでの浮動小数点計算における例外チェックを行うための浮動小数点例外チェックユニット（EC1及びEC2 ）１０９及び１１０と；ＷステージでMA0 １０７、MA1 １０８、EC1 １０９及びEC2 １１０の出力を受ける４つの書き込みポートと、Ｅステージでオペランドデータ供給線１１２～１１９を介してALU0１０３、ALU1１０４、FADD１０５及びFMUL１０６へオペランドデータを供給するための８つの読み込みポートとからなる１２のポートを有するマルチポートレジスタファイル１１１と、を備えている。

【００２２】この図１の構成においては、MA0 １０７及びMA1 １０８によってページフォールトやオーバーフローなどの算術演算例外トラップが発生し、又、EC1 １０９及びEC2 １１０によって浮動小数点演算例外トラップが発生する。

【００２３】このような例外トラップに対処するために、このプロセッサシステムは更に、トラップ発生の原因を格納するためのトラップ原因レジスタ１３０と；トラップによって実行が遮断されている命令中で最も小さいアドレスを有する命令のアドレスを格納するためのアボートアドレスレジスタ１３１と；トラップ発生を起こした命令のアドレスを格納するためのトラップアドレスレジスタ１３２と；トラップ要求信号線１４３～１４６を介して送られる、MA0 １０７、MA1 １０８、EC1 １０９及びEC2 １１０からのトラップ原因を受け、それに応じてトラップ信号をトラップ信号線１３４を介してアサートし、一方、信号線１４０、１４１及び１４２を介してトラップ原因レジスタ１３０、アボートアドレスレジスタ１３１、トラップアドレスレジスタ１３２への入力を適宜発生させるトラップ制御ユニット１３３と、を備えている。

【００２４】トラップ信号線１３４におけるトラップ信号は、命令発行ユニット１０２、ALU0１０３、ALU1１０４、FADD１０５、FMUL１０６、MA0 １０７、MA1 １０８、EC1 １０９及びEC2 １１０へ各々送られる。トラップ制御ユニット１３３からのトラップ信号に応じて、実行無効化フラグが各部に立てられ、その後のパイプラインステージで命令の処理をアボートする一方、命令発行ユニット１０２は予め定められたトラップ処理ルーチンに関する命令フェッチを開始し、このトラップ処理ルーチンにおいて、トラップ原因レジスタ１３０、アボートアドレスレジスタ１３１及びトラップアドレスレジスタ１３２に各々格納されているトラップ原因、アボートアドレス及びトラップアドレスが使用される。

【００２５】更に詳細には、トラップ制御ユニット１３３は図２に示されるような構成を有している。即ち、トラップ制御ユニット１３３は更に、Ｍステージで現在実行される命令のアドレスの下位２ビットを除くことによって得られるワードアドレスの共通部分を格納するＭステージプログラムカウンタ１（MPC ）１５１と；Ｍステージで現在実行される命令のうち最も小さいアドレスを有する命令のアドレスの下位２ビットによって示されるものであってMPC １５１のエントリと組み合わされてアボートアドレスレジスタ１３１に供給されるアボートアドレスを発生することになるワードアドレスの個別部分を格納するＭステージプログラムカウンタ２（mpc ）１５２と；ＭステージでMA0 １０７、MA1 １０８、EC1 １０９及びEC2 １１０によって現在実行される命令のアドレスの下位２ビットによって示されるワードアドレスの個別部分を格納するためのＭステージサブプログラムカウンタ（submpc1，submpc2，submpc3 及びsubmpc4 ）１５３、１５４、１５５及び１５６と；Ｍステージサブプ

ログラムカウンタ１５３～１５６の中で最小のエントリを、MPC １５１のエントリと組み合わされてトラップアドレスレジスタ１３２に供給されるトラップアドレス１４２を発生するための出力１４７として出力し、最小のエントリを有するMステージサブプログラムカウンタ１５３～１５６に対応するトラップ要求信号線１４３～１４６のうちの１つを介して送られるトラップ原因をトラップ原因レジスタ１３０に供給されるトラップ原因１４０として出力するトラップデータ生成ユニット１５７と；トラップ原因をトラップ要求信号線１４３～１４６のいずれか１つから受けたときにアサートされるトラップ信号をトラップ信号線１３４に対して発生するトラップ信号生成ユニット１５８と、を備えている。

【００２６】図３は、図１のプロセッサシステムにおけるパイプライン処理において、前記図２１のプログラムが実行された時に「load」命令でページフォールトが起こった場合の進行状況をしめすものであり、斜線部分はアボートされた命令を示す。図３に示すように、図１の第１実施例においては、プログラム実行の進行中にn+2 番目の「load」命令の実行によりトラップが発生したとき、トラップを引き起こしているn+2 番目の命令と同時にフェッチされた命令番号がn からn+3 の命令全てがアボートされる。図２２に示される従来の場合と比較すると、アボートされる命令の数が従来の場合の方が少ないので従来の場合の方が効率的であると思われるかも知れないが、既に述べたように従来の場合は適切なトラップ信号を決定する時間が必要なために、従来の場合の方がこの第１実施例よりサイクル時間が長くなり、実際にはこの第１実施例の方が効率が高くなる。

【００２７】より詳しく説明すると、図３に示されるパイプライン処理において、n 番目、n+1 番目、n+2 番目及びn+3 番目の命令全てがサイクルC+3 でMステージに入いり、n 番目の「fadd」処理についてのMステージ処理がEC1 １０９で行われ、n+1 番目の「add 」処理についてのMステージ処理がMA0 １０７で行われ、n+2 番目の「load」処理についてのMステージ処理がMA1 １０８で行われ、n+3 番目の「fmul」処理についてのMステージ処理がEC2 １１０で行われる。このサイクルC+3 において、ページフォールトがMA1 １０８で検出されたとき、MA1 １０８はトラップ要求信号線１４４を介してトラップ制御ユニット１３３にページフォールトトラップの発生を知らせる。ここで、このサイクルC+3 において、MPC １５１は、Mステージで現在実行されているn番目～n+3 番目の命令の下位２ビットを除いたワードアドレスを格納し、mpc １５２は、Mステージで現在実行されている命令中最小のアドレスを有するn 番目の命令の下位２ビットを示すワードアドレスを格納しており、この場合これは０である。一方、submpc1 １５３、submpc2 １５４、submpc3 １５５及びsubmpc4 １５６は、MA0 １０７、MA1 １０８、EC1 １０９及びEC2 １１０の各々で現在実行されているn+1 番目、n+2 番目、n番目及びn+3 番目の命令の下位２ビットを各々示すワードアドレスを格納している。

【００２８】トラップ信号生成ユニット１５８は、トラップ要求信号線１４３～１４６のいずれかにトラップ要求があるとき、トラップ信号線１３４のトラップ信号をアサートする。トラップ信号線１３４におけるアサートされたトラップ信号は、命令発行ユニット１０２、ALU0 １０３、ALU1１０４、MA0 １０７、MA1 １０８、FADD１０５、FMUL１０６、EC1 １０９及びEC2 １１０に供給されるので、これらのユニットにおける処理がアボートされると同時に、命令発行ユニット１０２は予め定められたトラップ処理ルーチンに関する命令フェッチを始める。

【００２９】ここで、トラップデータ生成ユニット１５７がトラップ要求信号線１４４にトラップ要求を検出すると、このトラップ要求信号線１４４に対応するsubmpc2 １５４のエントリが、信号線１４７に出力され、それがMPC １５１のエントリと組み合わされてトラップアドレス１４２、この場合n+2 番目の命令のアドレス、を生成し、このn+2 番目の命令のアドレスはトラップアドレス信号線１４２を介してトラップアドレスレジスタ１３２に格納される。

【００３０】他方、MPC １５１のエントリをmpc １５２のエントリと組み合わせることによりアボートアドレス１４１が得られ、この場合のアボートアドレスであるn番目の命令のアドレスはアボートアドレス信号線１４１を介してアボートアドレスレジスタ１３１に格納される。

【００３１】更に、トラップ要求を現在出しているトラップ要求信号線に付随するMステージプログラムカウンタ中で最小のアドレスを格納するsubmpc1 １５３、submpc2１５４、submpc3 １５５及びsubmpc4 １５６の一つに対応するトラップ要求信号線１４３～１４６の１つにおける信号がトラップ原因信号線１４０へ出力され、トラップ原因レジスタ１３０に格納される。この場合、トラップ要求は、トラップ要求信号線１４４のみから出されいるので、トラップ要求信号線１４４の信号がトラップ原因信号線１４０へ出力されトラップ原因レジスタ１３０に格納されることになる。

【００３２】ここでは、n 番目からn+3 番目の命令全てが同時にフェッチされる場合を説明しているが、４つの命令間にデータの依存性があるためにmpc のエントリが０でない場合がありうる。又、２個以上のトラップ要求がトラップ要求信号線１４３～１４６において検出された場合には、submpc1 １５３、submpc2 １５４、submpc3 １５５及びsubmpc4 １５６の内でMステージで現在実行される命令中最小のアドレスを格納するもののエントリが信号線１４７に出力される。

【００３３】この実施例においては、トラップを引き起

こす命令とアボートされる命令は異なるので、アボートアドレスレジスタ１３１とトラップアドレスレジスタ１３２の両方が必要である。

【００３４】又、この実施例においては、トラップ要求信号線１４３～１４６のいずれであっても少なくとも１つトラップ要求があればすぐにトラップ信号はアサートされるのでトラップ信号線におけるエントリは非常に速く決めることができる。他方、トラップ原因レジスタ１３０に格納されるトラップ原因１４０及びトラップアドレスレジスタ１３２に格納されるトラップアドレス１４２は、Mステージサブプログラムカウンタ１５３～１５６におけるエントリを比較した結果として決定されるため、これらはずっと後にならないと決められない。しかし、一般的に言って、メモリアクセスをアボートするためにトラップ信号線におけるエントリが速く決められる必要はあるが、レジスタに格納されるトラップデータはずっと遅く決められても構わないので、この実施例においてはプロセッサシステムのサイクル時間を長くする必要は生じない。

【００３５】従って、この第１実施例によると、システムにおけるクロック周波数の低下を防げるように、サイクル時間を増加させることなく機能できるトラップ制御機能を組み込んだスーパースカラプロセッサ等の並列処理型プロセッサシステムを提供することが可能となる。

【００３６】次に、図４から図６を参照して、本発明に係る並列処理型プロセッサシステムの第２の実施例について詳細に説明する。

【００３７】この第２実施例は、上述の第１実施例の応用であり、以下の理由から、第１実施例のトラップ制御機能に加え、更に、ストール制御機能を組み込んだものである。

【００３８】即ち、第１実施例のシステムを変形して、図４に示すように、システムが更に、バス線を介して命令キャッシュメモリ１０１A及び２ポートデータキャッシュメモリ１２５Aに接続されるメインメモリ１６１とＩ／Ｏ装置１６２とを含むようにした場合、トラップ制御機能だけではトラブルが起きる場合がある。トラブルの原因は、ここでは、メモリアドレスにマップされる複数のＩ／Ｏレジスタを通常含んでいるＩ／Ｏ装置１６２である。このようなＩ／Ｏレジスタへのアクセスは、Ｉ／Ｏレジスタのアドレスを特定するメモリアクセス命令と同一の命令によってなされ、キャッシュは通常この目的には使用されない。Ｉ／Ｏレジスタは、コマンド及びパラメータをＩ／Ｏ装置１６２に設定するためとＩ／Ｏレジスタのステータスをプロセッサ側に示すためとに用いられ、例えばＩ／Ｏ装置１６２のステータスを読み込むためのアクセスに応じて、プロセッサ側でＩ／Ｏ装置１６２のステータスを読み込んだときにステータスレジスタを消去するような形で、Ｉ／Ｏレジスタの内部状態を変えてしまうタイプのＩ／Ｏレジスタがある。

【００３９】そのようなＩ／Ｏ装置１６２を使用する場合、上述の第１実施例のトラップ制御機能のみを有するシステムでは、以下のような問題に直面することになる。即ち、このシステムにおいては、２つのメモリアクセス命令が同時実行可能なような２ポートデータキャッシュメモリ１２５Aについて、２つのアクセスユニット１０７及び１０８が設けられている。従って、メモリアクセスユニット１０７及び１０８において、２つのＩ／Ｏアクセス命令がMステージに同時に到達する場合がある。しかし、Ｉ／Ｏアクセスの目的のためのラインはバス線１６０１つのみしかないので、２つのＩ／Ｏアクセス命令を同時に処理することは不可能である。この結果、２つのＩ／Ｏアクセス命令のうちの最初の１つの処理については２番目のＩ／Ｏアクセス命令の処理が始まる前にＩ／Ｏ装置１６２の側で完了していなければならない。

【００４０】このような状況において、２番目のＩ／Ｏアクセス命令の処理でバスエラーなどの例外が起きる可能性がある。そのような場合、もしシステムが上記第１実施例のトラップ制御機能しか用いていないと、両方のＩ／Ｏアクセス命令がアボートされることになる。従って、適切なトラップ処理ルーチンを行った後に元のプログラムを再度実行するとき、Ｉ／Ｏ装置１６２の側においては最初のＩ／Ｏアクセス命令は終わったと見なされているにも関わらず、これがもう一度実行されることになる。この場合、最初のＩ／Ｏアクセス命令が偶然に、ステータスレジスタを読む命令であると、ステータスレジスタはトラップ発生の前に既に一度読まれているので、このステータスレジスタの内容は既に消去されており、最初のＩ／Ｏアクセス命令を再度実行するときにはもはや正しいものではなくなってしまっているためトラブルが生じる。

【００４１】上記のトラブルは基本的に、Ｄステージでは検出できない同一演算リソースの使用に関する要求の競合がその後のパイプラインステージで起こり得ることによる。それ故、もしＤステージでこれらの２つのメモリアクセス命令が実際に２つのＩ／Ｏアクセス命令であることを検出できれば、これらの２つのメモリアクセス命令をプロセッサに同時に供給しないことによってトラブルは避けることが出来るが、その反面、Ｉ／Ｏアクセス命令のためだけに特別な命令を使用することを要求することになる。

【００４２】図４に示される第２実施例の構成においては、この問題を、以下のように、そのような特別なＩ／Ｏアクセス命令を使用せずに解決するようにしている。

【００４３】まず第１に、この第２実施例においては、命令がALU0１０３及びALU1１０４へ同時に供給されることになるとき、最初に実行されるべき、より小さいアドレスを有する命令がALU0１０３へ供給されるように、命令発行ユニット１０２AがALU0１０３及びALU1１０４へ

の命令供給を制御する。従って、メモリアクセス命令がMA0 １０７及びMA1 １０８へ同時に届いたときには、MA0 １０７が常に先に実行されるべきメモリアクセス命令を有していることになる。

【００４４】第２に、図４の構成は、ストール要求信号がMA0 １０７、MA1 １０８、EC1 １０９及びEC2 １１０からストール要求信号線１７０、７１、７２及び１７３を介して各々供給されるストール制御ユニット１６３を更に含んでおり、このストール制御ユニット１６３はstall1信号１８０、stall2信号１８１及びstallv1 信号１８２を命令発行ユニット１０２Ａ、ALU0１０３、ALU1１０４、FADD１０５、FMUL１０６、MA0 １０７、MA1 １０８、EC1 １０９、EC2 １１０及びトラップ制御ユニット１３３へ出力して、以下に記載するような適当なストール制御を行う。

【００４５】stall1信号１８０は、MA0 １０７、MA1 １０８、EC1 １０９及びEC2 １１０のいずれかからのストール要求があるときアサートされ、stall2信号１８１は、MA0 １０７、MA1 １０８、EC1 １０９及びEC2 １１０のいずれかからのストール要求があるときアサートされ、一方stallv1 信号１８２は、Mステージの処理がMA1 １０８で現在実行されている命令のアドレスの下位２ビットを示す。

【００４６】これらのstall1、stall2及びstallv1 信号１８０～１８２の値に従って、このシステムにおけるパイプライン処理は以下のように制御される。

【００４７】（1）ALU0１０３及びMA0 １０７のパイプライン＜１０３、１０７＞：

(a) Mステージ及びWステージ：stall2信号１８１がニゲートであれば、stall1信号１８０に関係なくパイプライン処理が行われる。

【００４８】(b) Eステージ：stall1信号１８０がニゲートであれば、パイプライン処理が行われる。

【００４９】（2）ALU1１０４及びMA1 １０８のパイプライン＜１０４、１０８＞：stall1信号１８０がニゲートであれば、パイプライン処理が行われる。

【００５０】（3）FADD１０５及びEC1 １０９のパイプライン＜１０５、１０９＞及びFMUL１０６及びEC2 １１０のパイプライン＜１０６、１１０＞：

(a) Mステージ及びWステージ：stall1信号１８０がニゲートであれば、stall1信号１８０に関係なくパイプライン処理が行われる。

【００５１】パイプライン処理は又、stall1信号１８０がアサートでありstall2信号がニゲートであって、stallv1 信号１８２が、EC1 １０９及びEC2 １１０の各々においてMステージの処理が現在行われている命令の下位２ビットを格納するsubmpc3１５５及びsubmpc4 １５６の各々より大きい値を示している場合に行われる。

【００５２】(b) Eステージ：stall1信号１８０がニゲートであれば、パイプライン処理が行われる。

【００５３】又、この第２実施例においては、mpc １５２に格納される値はこれらstall1、stall2及びstallv1 信号１８０～１８２の値に従って以下の様に決定される。即ち、stall1信号１８０がアサートされないとき、mpc １５２には、Eステージで実行された命令中の最小アドレスがロードされ、stall1信号１８０及びstall2信号１８１の両方がアサートされるときは、mpc １５２は前の値を維持するが、stall1信号１８０がアサートされるがstall2信号１８１信号はアサートされないときには、mpc １５２にはstallv1 信号１８２によって示される値がロードされる。

【００５４】従って、この第２実施例では、MA1 １０８だけからストール要求があれば、パイプライン＜１０３、１０７＞の処理が完了し、現在処理されている命令のアドレスがMA1 １０８での命令のアドレスより小さいときのみ、パイプライン＜１０５、１０９＞と＜１０６、１１０＞の各々の処理が完了する。

【００５５】この結果、第１及び第２Ｉ／Ｏアクセス命令が同時にMA0 １０７及びMA1 １０８に各々届いたとき、MA1 １０８はストール要求をストール制御ユニット１６３に出力するが、MA0 １０７は、１クロックサイクル以内でＩ／Ｏアクセス処理を完了させるのが不可能でない限り、ストール要求を出力しない。そして、EC1 １０９及びEC2 １１０もストール要求を出力しない時、stall1信号１８０のみがアサートされ、stall2信号１８２はアサートされない。そのような場合、パイプライン＜３０３、１０７＞は処理が進められ、mpc １５２には、stallv1 信号１８２の値がロードされて，システムの状態が２番目のＩ／Ｏアクセス命令より前の命令が実行完了している状態となる。

【００５６】それ故、２番目のＩ／Ｏアクセス命令について例外が発生したときでも、最初のＩ／Ｏアクセス命令は再度実行されない。他方、パイプライン＜１０４、１０８＞の処理と、２番目のＩ／Ｏアクセス命令より大きいアドレスを有する命令についてのパイプラインの処理は、stall1信号１８０がニゲートになる迄ストールされる。

【００５７】１クロックサイクル以内にＩ／Ｏアクセス処理を完了させるのが不可能であるためにMA0 １０７がストール要求を出力するときには、最初のＩ／Ｏアクセス命令が完了してstall2信号１８１がニゲートになるまで、stall1信号及びstall2信号１８２がアサートされるので、全てのパイプラインはストールされる。

【００５８】図６は、図４のプロセッサシステムにおけるパイプライン処理において、図５のプログラムを実行した時にn+2 番目の「ロード」命令でバスエラーが起こった場合の進行状況を示すものである。図６に示されるように、この図４の第２実施例においては、ストール要求がサイクルC+3 においてn+2 番目の命令の実行によって発生するとき、EC1 １０９で行われるn 番目の「fad

d」処理についてのMステージ処理及びMA0 １０７で行われるn+1 番目の「add 」処理についてのMステージ処理は、次のサイクルC+4 で終了するが、MA1 １０８でのn+2 番目の「load」操作についてのMステージ処理及びEC2 １１０でのn+3 番目の「fmul」操作についてのMステージ処理及びは次のサイクルC+4 でストールされ、その後のサイクルC+5 迄終了しない。一方、n+2　番目の命令より大きいアドレスをもった後続の命令も全てサイクルC+4 でストールされる。

【００５９】ここで、命令の実行中に例外が起こる可能性を否定できないために命令の処理がストールされた後で、命令の実行中に実際に例外が発生したときには、命令の処理はアボートされる。さもなくば、命令実行中に実際には例外が起こらなかったので命令の処理が再開される。

【００６０】従って、この第２実施例によると、システムにおけるクロック周波数の低下を防げるように、サイクル時間を増加させることなく機能できるトラップとストールの制御機能を組み込んだスーパースカラプロセッサ等の並列処理型プロセッサシステムを提供することが可能となる。

【００６１】次に図７を参照して、本発明に係る並列処理型プロセッサシステムの第３の実施例を詳細に説明する。

【００６２】この第３実施例は、より一般的な設定における上記第２実施例のストール制御機能の一般化を行ったものである。

【００６３】この第３実施例においては、図７に示されるように、プロセッサシステムは、命令を格納する命令キャッシュメモリ（I-cache ）２０１と；Ｆステージで I-cache ２０１から４ワードバウンダリの４つの命令を同時にフェッチし、Ｄステージで４つのフェッチされた命令間のデータ依存関係及び制御依存関係を考慮し、Ｅステージで命令供給線２２０、２２１、２２２及び２２３を介して実行可能な命令を供給する命令発行ユニット２０２と；命令供給線２２０及び２２１から供給される命令に従って、Ｅステージで算術論理演算及びメモリアドレス計算を実行する論理演算ユニット（ALU0及びALU1）２０３及び２０４と；ALU0２０３及びALU1２０４からのコマンドに従ってＥステージで整数の乗除算を行うための整数乗除算器２０５と；ALU0２０３及びALU1２０４からのアクセスされるデータを格納するためのデータキャッシュメモリ（D-cache ）２０６と；命令供給線２２２から供給される命令に従ってＥステージで浮動小数点の加減算を行うための浮動小数点加算器（FADD）２０７と；命令供給線２２３から供給される命令に従ってＥステージで浮動小数点の乗算を行うための浮動小数点乗算器（FMUL）２０８と；FMUL２０８からのコマンドに従ってＥステージで浮動小数点の除算を行うための浮動小数点除算器（FDIV）２０９と；命令発行ユニット２０２で発行される命令を特定するための命令アドレス生成ユニット２１０と；後に説明するトラップ制御及びストール制御を行うための制御ユニット２１１と；ALU0２０３、ALU1２０４の出力を格納するための整数レジスタファイル２１２と；FADD２０７及びFMUL２０８の出力を格納するための浮動小数点レジスタファイル２１３とを備えている。

【００６４】上述の図４の第２実施例と同様に、図７のプロセッサシステムは、更に、I-cache ２０１及びD-cache ２０６にキャッシュされるデータを格納するためのメインメモリ２１４と；Ｉ／Ｏレジスタを含むＩ／Ｏ装置２１５と；メインメモリ２１４及びＩ／Ｏ装置２１５をI-cache ２０１及びD-cache に接続させるバス線２１６とを有している。

【００６５】加えて、図７のプロセッサシステムには、更に、I-cache ２０１とバス線２１６との間に設けられるプレデコーダ２１７と；命令発行ユニット２０２に接続されているレジスタスコアボード回路２１８とが組み込まれている。これらについては以下に詳細に説明する。

【００６６】まず、この第３実施例におけるトラップとストールの制御処理について説明する。

【００６７】一般に、並列処理型プロセッサシステムは、マシン語命令の実行の時に同じリソースについての競合する要求が発生するのを回避する機構、及び、マシン語命令間の実行順序の正当性を保持する機構を有する必要がある。ここで、実行順序の正当性とは、データ依存関係及び制御依存関係のコンシステンシを意味する。

【００６８】データ依存関係のコンシステンシを維持するためには、実行順序をＤ→Ｓ関係、Ｓ→Ｄ関係及びＤ→Ｄ関係の内の１つに維持する必要がある。ここで、Ｄ→Ｓ関係は、先に実行すべき命令の結果を格納するリソースが、後で実行すべき命令に用いられるソースデータが読みだされるリソースと同一である関係を示す。Ｓ→Ｄ関係は、先に実行すべき命令の結果を格納するリソースが、後で実行すべき命令の結果を格納するためのリソースと同一である関係を示す。Ｄ→Ｄ関係は、先に実行すべき命令に用いられるソースデータが読みだされるリソースが、後で実行すべき命令に用いられるソースデータが読みだされるリソースと同一であることを示す。この図７の第３実施例においては、データ格納リソースが、レジスタ及びメモリの２つの態様で存在し、両者間のデータ依存関係のコンシステンシを保つ必要がある。

【００６９】制御依存関係とは、先の分岐命令とその後の命令との間の関係である。従来のＶＬＩＷ型プロセッサにおける分岐命令については、この制御依存関係のコンシステンシがコンパイラによって保たれている。しかし、この第３実施例の並列処理型プロセッサシステムでは、ユーザープログラムのオブジェクトコンパチビリティをもたせる必要があるので、制御依存関係のコンシス

テンシの保持はハードウェアにおいて実現させる必要がある。

【００７０】この第３実施例においては、プロセッサシステムは、４ワードバウンダリの４つの命令を同時にフェッチし、その４つの命令間でインオーダーな順序で同時に発行されできる命令を各演算ユニットに発行して、インオーダーな順序で命令の実行が完了する。

【００７１】この第３実施例においては、同じリソースを使用する競合する要求が発生するのを回避しデータ依存関係及び制御依存関係のコンシステンシを保持するためのハードウェアは、２つの機構を含んでいる。第１の部分は、命令発行ユニット２０２によって実現される命令発行機構であり、第２の部分は、制御ユニット２１１によって実現されるストール機構である。

【００７２】命令発行機構を実現するために、命令発行ユニット２０２には、同じリソースを使用する競合する要求の発生を検出するためのプレデコーダ２１７が備えられており、これはキャッシュへのリフィル時（又はキャッシュスルー命令フェッチの時は命令フェッチの時）に同時にフェッチされた４つの命令の使用リソースと、同時にフェッチされた命令のうちで最小のアドレスを有する命令について競合があるかどうかとのマークを付ける。命令発行ユニット２０２は又、D→S関係及びD→D関係を検出するためのレジスタスコアボード回路２１８も備えており、プレデコーダ２１７によってマークされる競合を回避しつつ、レジスタスコアボード回路２１８によって検出されるデータ依存関係及び制御依存関係においてD→S及びD→D関係を保つことにより、Dステージでの命令発行処理をインオーダーな順序で行うことができる。S→D関係については、ソースレジスタの読み出し後に、命令発行をインオーダーな順序で行いインオーダーな順序で実行が完了することにより保たれる。

【００７３】更に、この第３実施例においては、２つのメモリアクセス命令を同時に実行することが可能であるために、メモリリソースについてのデータ依存関係と同様、メモリリソースの競合関係のコンシステンシも保つ必要がある。命令発行ユニット２０２はDステージより前で処理を行うので、命令発行ユニット２０２はこれらの関係がないと仮定して処理を行っており、これらの関係が存在するかどうかを正確に確認することができない。これらの関係の存在はMステージになるまで正確には決められず、関係の存在の可能性を決めることは可能であっても、命令供給処理をそのような可能性に基づいて過剰に制御してしまうと、パーフォーマンスが著しく制限されてしまうことになる。

【００７４】ストール機構においては、実行完了はインオーダーな順序に保たれ、命令発行機構で既に考慮されたもの以外のケース及び命令発行処理が速すぎて命令フェッチ処理が追いつけなくなった場合についての制御を行う。

【００７５】インオーダーな順序での実行完了においては、Dステージから同時発行された命令は同時に実行完了するという基本原理に基づいた命令の実行完了が達成される。但し、ALU１２０４によってMステージで実行されるていメモリアクセス命令Xによるメモリアクセス処理が１サイクル時間内に終了できない時には、上記第２実施例と同様に、命令Xによるストール要求に関わらず、このメモリアクセス命令Xより小さいアドレスを有する命令を実行完了し、実行完了した命令の数だけmpcが更新される。この機能を実行完了ステージにおけるグループ機能と呼ぶ。この第３実施例においては、ALU０２０３及びALU１２０４で命令を同時に実行するとき、命令発行ユニット２０２は、ALU０２０３が常により小さいアドレスを有する命令を実行するような制御を行うので、インオーダーな実行完了が保証される。この命令発行ユニット２０２による制御によって、Dステージでは検出することができないメモリリソースの使用に関する競合する要求の発生によるデッドロックをMステージにおいて防ぐことができる。

【００７６】この様な、ストール要求を発している命令より小さいアドレスを有する命令のみについてパイプライン処理を完了するための制御は、他の状況、例えば、ALU０２０３でのMステージ処理が１サイクルで終了できないような場合や、FADD２０７又はFMUL２０８のE2ステージで実行される命令についてトラップの可能性が否定できないためにストール要求がアサートされている場合等にも同様に適用することが可能である。

【００７７】命令発行ユニット２０２は、メモリ（キャッシュメモリを含む）に関する限り、同じリソースの使用に関する競合する要求の発生あるいはデータ依存関係を考慮しない。この第３実施例のプロセッサシステムはRISCと同じように、いわゆる「load，store 」アーキテクチュアを採用するので、同じリソースの使用に関する競合する要求の発生の防止及びデータ依存関係の保持を確保するためには、「load」及び「store 」命令のみを考慮すれば充分である。D-cache ２０６にはALU０２０３及びALU１２０４用の専用ポートがあるので、ALU０２０３及びALU１２０４の１つだけがD-cache ２０６にアクセスしている限りは、リソースの競合は起こらない。故に、「load」及び「store 」命令は外部メモリに同時にアクセスする際、データ依存関係はこの２つの命令が同じアドレスにアクセスしようとしている場合にのみ存在することになる。

【００７８】メモリリソース競合は実行完了ステージにおける命令グループ機能によって解決される。メモリについてのデータ依存関係には、「リードアフターライト」、「ライトアフターリード」及び「ライトアフターライト」があり、そのコンシステンシはキャッシュメモリの側で保たれる。

【００７９】次に図８から図１８を参照して、図７の第３実施例におけるストール制御を詳細に説明する。

【００８０】この第３実施例においては、ストールが生じる場合は以下のように要約することができる。

【００８１】MObusy
M1busy
Imis（I-cache miss）
FRbusy（FPUFA レジスタ書き込み競合）
FAexch（FADD例外チェック）
FMexch（FMUL例外チェック）
FDexch（FMUL例外チェック）
Fstall（強制ストール）
これらの場合の各々に対応するストール要求信号がアサートされる条件は以下のようなものである。

【００８２】MObusy、M1busy：これらのタイプのストール要求信号は、メモリアクセス処理（キャッシュ及びＩ／Ｏに関するアクセス処理を含む）がMステージでt 番目のサイクルに行われ、このメモリアクセス処理がこのt 番目のサイクル中に完了できない時にアサートされる。

【００８３】Imis：このタイプのストール要求信号は、新しい命令フェッチ要求があるがその命令フェッチが不成功の時に、アサートされる。ストールがこのImisストール要求によって起こった場合の例を図８から図１２に示す。ここでは、Imisストール要求信号は、fpc に新しい値がロードされ「fpcen 」をアサートされてから１クロックサイクル後に命令が命令レジスタにフェッチされない時、実際にアサートされる。

【００８４】図８は命令フェッチ時に命令キャッシュミスによるストールが発生した場合を示している。

【００８５】図９は命令フェッチ時に命令キャッシュミスによるストールが発生した後、システム中の他の部分で別のストール要求が発生し、命令キャッシュミスによるストールが他の部分での別のストールより先に解消された場合を示している。

【００８６】図１０は命令キャッシュミスによるストールがシステムの他の部分での別のストールと同時に発生し、他の部分での別のストールの方が命令キャッシュミスによるストールよりも先に解消した場合を示している。

【００８７】図１１は命令フェッチ時の命令キャッシュミスによるストールがジャンプ命令のジャンプ先で発生した場合を示している。

【００８８】図１２は命令キャッシュミスによるストールに伴うキャッシュリフィル動作中にジャンプが起こった場合を示している。

【００８９】FRbusy：このタイプのストール要求信号は、FDIV２０９がパイプラインのE2ステージにあってFMUL２０８もこのE2ステージにある時にアサートされ、FDIV以外のパイプライン処理を１サイクルの間ストールすることによって、FMUL２０８とFDIV２０９間での浮動小数点レジスタファイル２１３への書き込みの競合を回避するようにするものである。

【００９０】FAexch：FADD２０７のE1ステージにおいて、トラップ発生の可能性が否定できないとき、FADD２０７におけるパイプライン処理は通常のF1型からF2型になる。この場合、FADD２０７での実行終了ステージはMステージになるため、FADD２０７のE2及びE3ステージの間他のユニットの実行完了ステージを遅延させて全てのユニットがFADD２０７のMステージと同時に実行完了ステージに達するようにするために、このFAexchストール要求信号をアサートすることによって他のユニットの処理をストールする。このようなこのFAexchストール要求信号によってストールが起こる状況が、図１３に示されており、ここでは「fadd」命令と同時にフェッチされた「Iadd」「fmul」命令の処理と、次のサイクルでフェッチされる「fadd」「Iadd」及び「fmul」命令の処理が「fadd」命令の処理のE2及びE3ステージ間にストールされる。

【００９１】FMexch：FMUL２０８のE1ステージにおいて、トラップ発生の可能性が否定できない場合、FMUL２０８でのパイプライン処理は、通常のF1型からF2型タイプになる。この場合、FMUL２０８での実行完了ステージはMステージになるので、他のユニットの実行終了ステージを２サイクル遅延させて全てのユニットがFMUL２０８のMステージと同時に実行完了ステージに達するようにするために、このFMexchストール要求信号をアサートすることによって他のユニットの処理をストールする。

【００９２】FDexch：FDIV２０９のE1ステージにおいて、トラップ発生の可能性が否定できない場合、FDIV２０９でのパイプライン処理は、通常のD1型からD2型になる。この場合、FDIV２０９での実行完了ステージはMステージになるので、他のユニットの実行完了ステージをFDIV２０９がE3ステージを通過する迄遅延させて全てのユニットがFDIV２０９のMステージと同時に実行完了ステージに達するようにするために、このFDexchストール要求信号をアサートすることによって他のユニットの処理をストールする。このようなこのFDexchストール要求信号によってストールが起こる場合が図１４に示されており、「fdiv」命令と同時にフェッチされた命令「iadd」及び「fadd」の処理は、命令「fdiv」がE3ステージを通過するまでストールされる。

【００９３】Fstall：このタイプのストール要求信号は、実行完了ステージ以前にキャッシュミスリカバリのような処理を除く全てのパイプラインの処理をロックさせるために、外部からアサートされる。

【００９４】図１５から図１７は、この第３実施例におけるストール制御のタイミングチャートの例を示す。

【００９５】図１５のタイミングチャートにおいては、N 番目のサイクルでALUO　２０３のMステージにおいて

キャッシュミスが検出されて、MObusy信号がアサートされる一方で、キャッシュミスリカバリ処理がスタートする。一方、FMUL２０８についてトラップ発生の可能性が否定されないので、FMexch信号がアサートされ、stall1信号及びstall2信号がアサートされる。。

【００９６】N+2 番目のサイクルにおいては、ALU0２０３についてキャッシュミスリカバリ処理が続けられる一方、FMUL２０８は例外を生じなかったのでFMexch信号はニゲートされる。しかし、stall1信号及びstall2信号はまだアサートされているので、FMUL２０８の実行は完了できない。

【００９７】N+4 番目のサイクルにおいては、ALU0２０３についてキャッシュミスリカバリ処理がこのサイクル中に完了できるので、MObusy信号がニゲートされる。このため、stall1信号及びstall2信号はニゲートされるので、他の全てのユニットでの実行が完了できるようになる。

【００９８】そして最後に、N+5 番目のサイクルにおいて、全ての命令の実行が完了されている。

【００９９】図１６のタイミングチャートにおいては、各処理ユニットでの命令のアドレスは、ALU0２０３＜ALU1２０４且つFMUL２０８＜ALU1２０４＜FADD２０７であると仮定している。この場合、キャッシュミスがN 番目のサイクルにおいてALU1２０４のMステージで検出されて、M1busy信号がアサートされる一方で、キャッシュミスリマバリ処理がスタートする。一方、トラップ発生の可能性がFMUL２０８について否定できないので、FMexch信号がアサートされ、stall1信号及びstall2信号がアサートされる。

【０１００】N+2 番目のサイクルにおいては、ALU1２０４についてキャッシュミスリカバリ処理が続けられる一方、FMUL２０８は例外を起こさなかったのでFMexch信号はニゲートされる。この時点で、stall2信号はニゲートされるのでALU0２０３及びFMUL２０８での実行は完了できるが、stall1信号がまだアサートされているので、FADD２０７の実行は完了できない。

【０１０１】N+4 番目のサイクルにおいては、ALU1２０４についてキャッシュミスリカバリ処理がこのサイクル中に完了できるので、M1busy信号がニゲートされる。このため、stall1信号はニゲートされるので、FADD２０７の実行が完了できるようになる。

【０１０２】そして最後に、N+5 番目のサイクルにおいて、全ての命令の実行が完了されている。

【０１０３】図１７のタイミングチャートにおいては、N 番目のサイクルにおいてALU1２０４のMステージでキャッシュミスが検出されて、M1busy信号がアサートされる一方でキャッシュミスリカバリ処理がスタートする。一方、トラップ発生の可能性がFMUL２０８について否定できないので、FMexch信号がアサートされ、stall1信号及びstall2信号がアサートされる。

【０１０４】N+2 番目のサイクルにおいては、ALU1２０４についてキャッシュミスリカバリ処理が完了し、M1busy信号がニゲートされ、同じN+2 番目のサイクル又はひとつ前のN+1 番目のサイクルにおいて、FMUL２０８が例外を起こさなかったのでFMexch信号がニゲートされる。この時点で、stall1信号及びstall2信号もニゲートされるので全ての命令の実行が完了できるようになる。

【０１０５】この第３実施例におけるstall1信号、stall2信号及びFRbusy信号を用いたストール制御処理に応じた各処理ステージでの各パイプラインの処理について図１８に示される表に要約する。

【０１０６】この第３実施例においては、命令実行において例外発生の可能性を否定できないとして命令の処理がストールされた後、命令の処理は、例外が実際に命令実行の際に起こったときにアボートされるか、さもなくば、命令の実行において例外が実際には起こらなかったときに命令が復帰する。図１８において、BUはプロセッサシステムの各処理ユニットに設けられた分岐ユニットを指すものである。

【０１０７】故に、この第３実施例によると、システムのクロック周波数の低下を防げるように、サイクル時間を増加することなく処理可能なトラップとストールの制御機能を組み込んだスーパースカラプロセッサ等の並列処理型プロセッサシステムを提供することが可能となる。

【０１０８】

【発明の効果】以上説明したように、本発明の並列処理計算機は、サイクル時間を増加させることなく、高速にトラップとストールの制御処理が可能なものであり、スーパースカラプロセッサ等の並列処理型プロセッサシステムにおいてシステムのクロック周波数を低下させることなくトラップとストールの制御をより効率的に行うことが可能となる。

【図面の簡単な説明】

【図１】本発明に係る並列処理型プロセッサシステムの第１実施例のブロック図である。

【図２】図１の並列処理型プロセッサシステムにおけるトラップ制御ユニットのブロック図である。

【図３】図２１のプログラムの実行時にトラップ要求が生じた場合の図１の並列処理型プロセッサシステムにおけるパイプライン処理の進行状況を示す図である。

【図４】本発明に係る並列処理型プロセッサシステムの第２実施例のブロック図である。

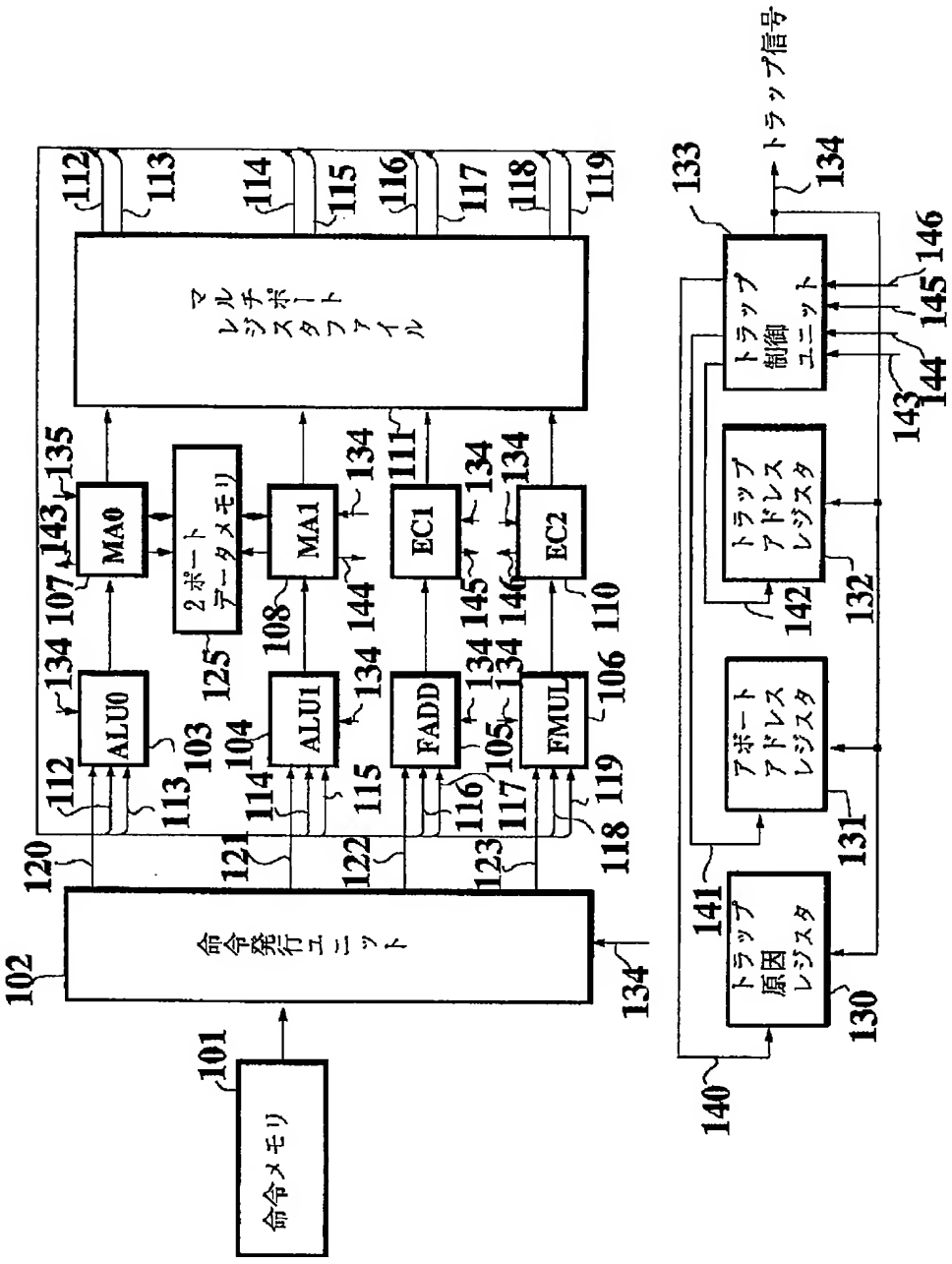
【図５】図４の並列処理型プロセッサシステムで実行されるプログラムの一例を示す図である。

【図６】図５のプログラムの実行時にトラップ要求が生じた場合の図４の並列処理型プロセッサシステムにおけるパイプライン処理の進行状況を示す図である。

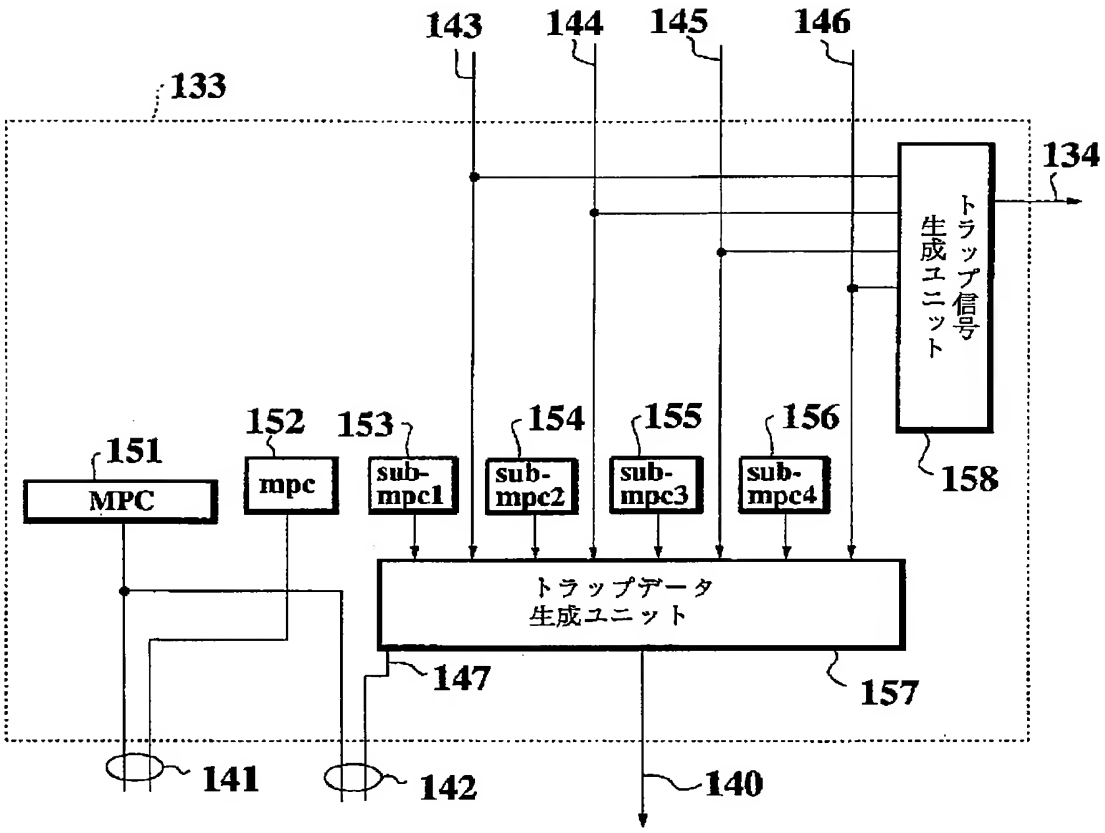
【図７】本発明に係る並列処理型プロセッサシステムの第３実施例のブロック図である。

【図８】図７の並列処理型プロセッサシステムにおける、命令キャッシュミスによるストールの一例を示す図である。
【図９】図７の並列処理型プロセッサシステムにおける、命令キャッシュミスによるストールの別の一例を示すタイミングチャートである。
【図１０】図７の並列処理型プロセッサシステムにおける、命令キャッシュミスによるストールの別の一例を示すタイミングチャートである。
【図１１】図７の並列処理型プロセッサシステムにおける、命令キャッシュミスによるストールの別の一例を示すタイミングチャートである。
【図１２】図７の並列処理型プロセッサシステムにおける、命令キャッシュミスによるストールの別の一例を示すタイミングチャートである。
【図１３】図７の並列処理型プロセッサシステムにおける、FAexchによるストールの一例を示すタイミングチャートである。
【図１４】図７の並列処理型プロセッサシステムにおける、FDexchによるストールの一例を示すタイミングチャートである。
【図１５】図７の並列処理型プロセッサシステムにおける、ストール制御処理の一例を示すタイミングチャートである。
【図１６】図７の並列処理型プロセッサシステムにおける、ストール制御処理の別の一例を示すタイミングチャートである。
【図１７】図７の並列処理型プロセッサシステムにおける、ストール制御処理の別の一例を示すタイミングチャートである。
【図１８】図７の並列処理型プロセッサシステムにおける、ストール制御処理に対する各パイプラインの処理態様をまとめた表である。
【図１９】従来のトラップ制御方法を用いた並列処理型プロセッサシステムのブロック図である。
【図２０】図１９の従来の並列処理型プロセッサシステムにおけるトラップ制御ユニットのブロック図である。
【図２１】並列処理型プロセッサシステムで実行されるプログラムの一例を示す図である。
【図２２】図２１のプログラムの実行時にトラップ要求が生じた場合の図１９の並列処理型プロセッサシステムにおけるパイプライン処理の進行状況を示す図である。
【符号の説明】
１０１　命令メモリ
１０１Ａ　命令キャッシュメモリ
１０２　命令発行ユニット
１０２Ａ　命令発行ユニット
１０３　算術論理演算ユニット
１０４　算術論理演算ユニット
１０５　浮動小数点加算器
１０６　浮動小数点乗算器
１０７　メモリアクセスユニット
１０８　メモリアクセスユニット
１０９　浮動小数点例外チェックユニット
１１０　浮動小数点例外チェックユニット
１１１　マルチポートレジスタファイル
１２５　２ポートデータメモリ
１２５Ａ　２ポートデータキャッシュメモリ
１３０　トラップ原因レジスタ
１３１　アボートアドレスレジスタ
１３２　トラップアドレスレジスタ
１３３　トラップ制御ユニット
１５１　Ｍステージプログラムカウンタ１
１５２　Ｍステージプログラムカウンタ２
１５３　Ｍステージサブプログラムカウンタ
１５４　Ｍステージサブプログラムカウンタ
１５５　Ｍステージサブプログラムカウンタ
１５６　Ｍステージサブプログラムカウンタ
１５７　トラップデータ生成ユニット
１５８　トラップ信号生成ユニット
１６０　バスライン
１６１　メインメモリ
１６２　Ｉ／Ｏ装置
１６３　ストール制御ユニット
２０１　命令キャッシュメモリ
２０２　命令発行ユニット
２０３　算術論理演算ユニット
２０４　算術論理演算ユニット
２０５　整数乗除算器
２０６　２ポートデータキャッシュメモリ
２０７　浮動小数点加算器
２０８　浮動小数点乗算器
２０９　浮動小数点除算器
２１０　命令アドレス生成ユニット
２１１　制御ユニット
２１２　整数レジスタファイル
２１３　浮動小数点レジスタファイル
２１４　メインメモリ
２１５　Ｉ／Ｏ装置
２１６　バスライン
２１７　プリデコーダ
２１８　レジスタスコアボード回路

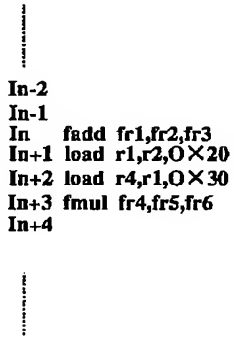
【図１】


命令メモリ
命令発行ユニット
マルチポートレジスタファイル
ALU0
ALU1
FADD
FMUL
MA0
MA1
EC1
EC2
2ポートデータメモリ
アボートアドレスレジスタ
トラップアドレスレジスタ
トラップ原因レジスタ
トラップ制御ユニット
トラップ信号

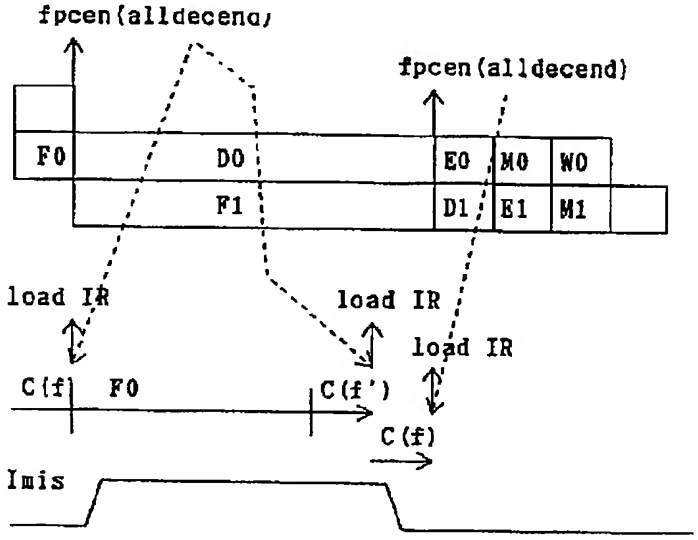
【図２】


143
144
145
146
133
134
トラップ信号
生成ユニット
152
153
154
155
156
151
MPC
mpc
sub-
mpc1
sub-
mpc2
sub-
mpc3
sub-
mpc4
158
トラップデータ
生成ユニット
147
157
141
142
140

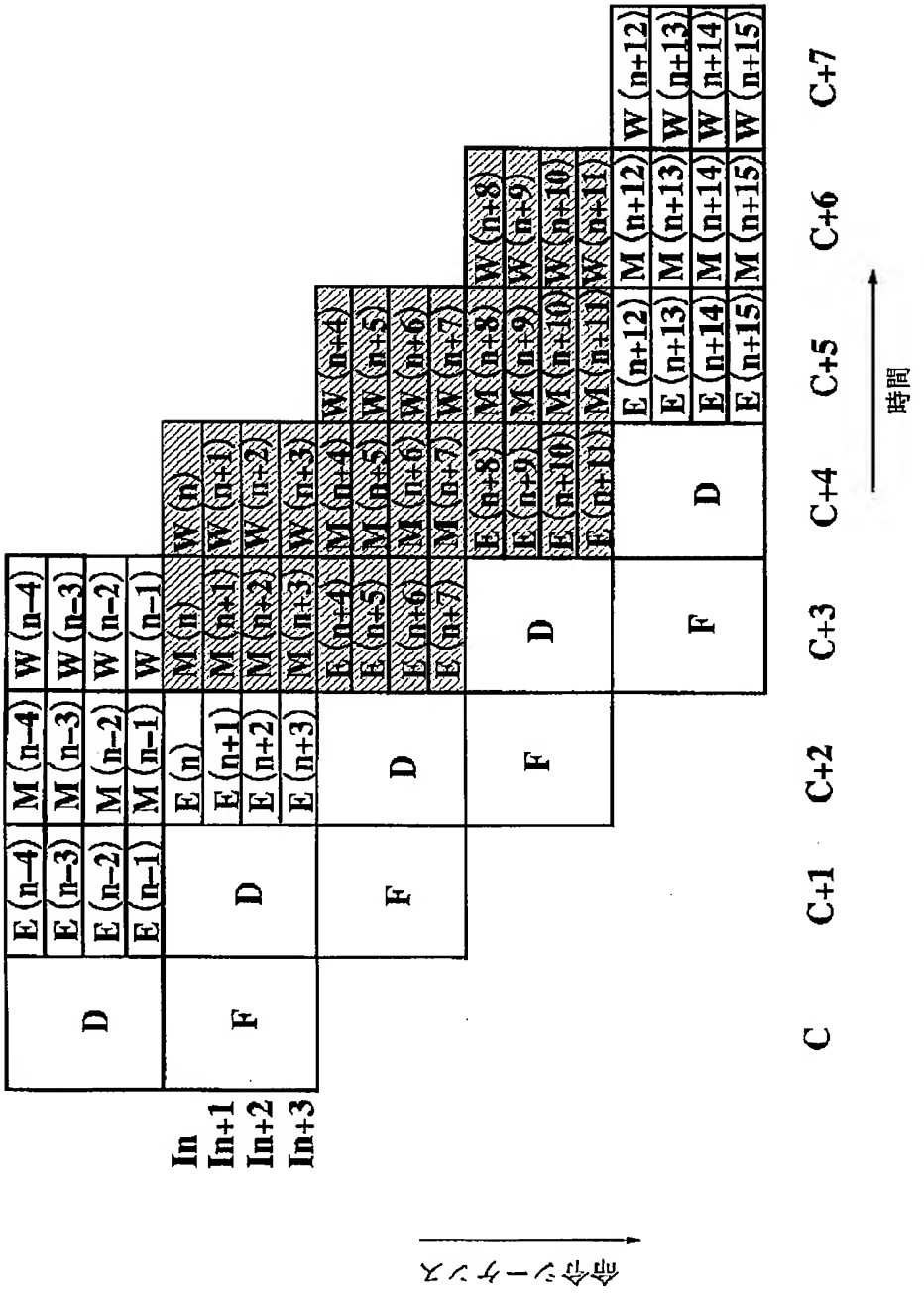
【図５】


In-2
In-1
In fadd fr1,fr2,fr3
In+1 load r1,r2,O×20
In+2 load r4,r1,O×30
In+3 fmul fr4,fr5,fr6
In+4

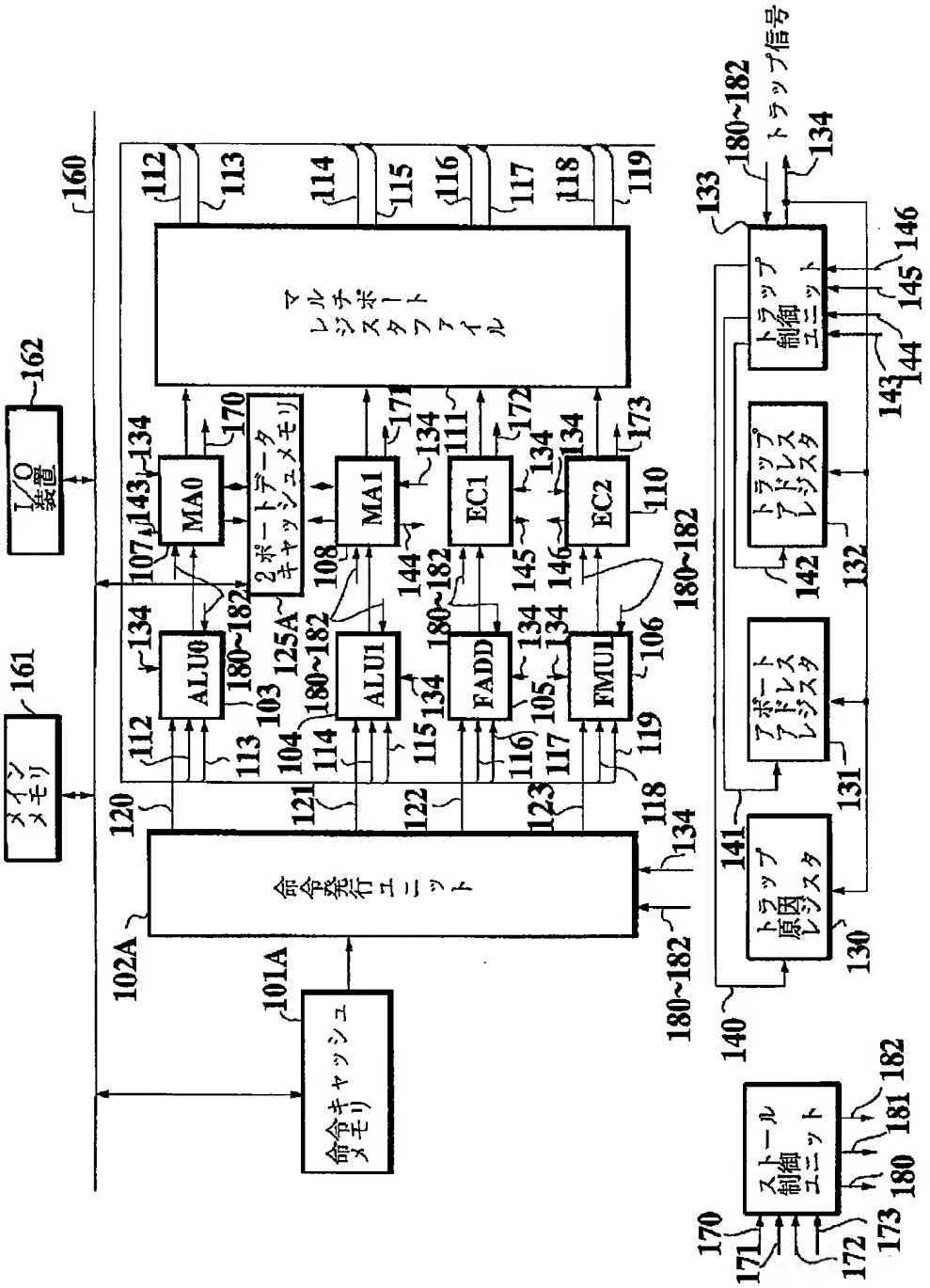
【図８】


fpcen(alldecend)
fpcen(alldecend)
F0
D0
E0
M0
W0
F1
D1
E1
M1
load IR
load IR
load IR
C(f)
F0
C(f')
C(f)
Imis

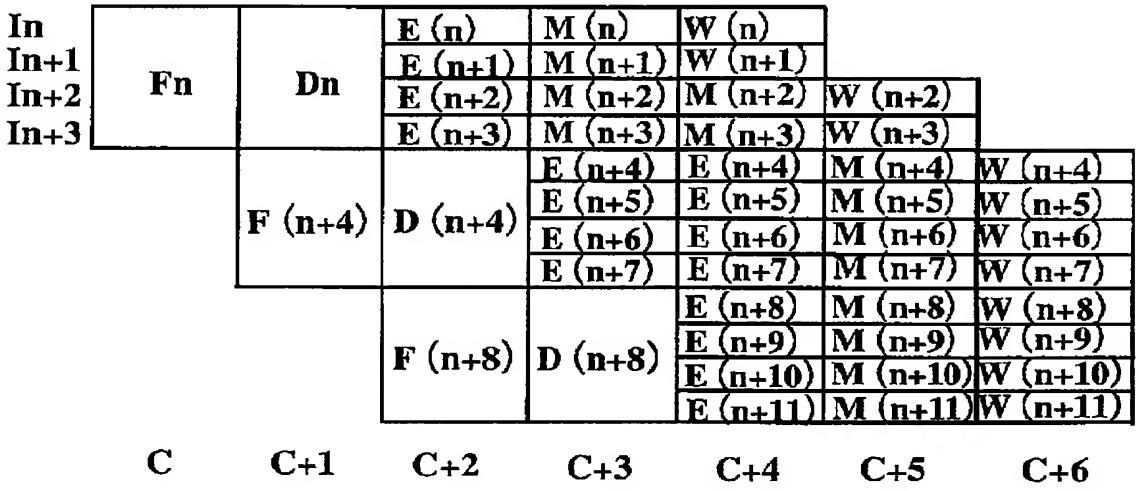
【図３】


C
C+1
C+2
C+3
C+4
C+5
C+6
C+7
時間
命令シーケンス
In
In+1
In+2
In+3
D
F
D
F
D
F
D
F
E (n-4)
E (n-3)
E (n-2)
E (n-1)
M (n-4)
M (n-3)
M (n-2)
M (n-1)
W (n-4)
W (n-3)
W (n-2)
W (n-1)
E (n)
E (n+1)
E (n+2)
E (n+3)
M (n)
M (n+1)
M (n+2)
M (n+3)
W (n)
W (n+1)
W (n+2)
W (n+3)
E (n+4)
E (n+5)
E (n+6)
E (n+7)
M (n+4)
M (n+5)
M (n+6)
M (n+7)
W (n+4)
W (n+5)
W (n+6)
W (n+7)
E (n+8)
E (n+9)
E (n+10)
E (n+11)
M (n+8)
M (n+9)
M (n+10)
M (n+11)
W (n+8)
W (n+9)
W (n+10)
W (n+11)
E (n+12)
E (n+13)
E (n+14)
E (n+15)
M (n+12)
M (n+13)
M (n+14)
M (n+15)
W (n+12)
W (n+13)
W (n+14)
W (n+15)

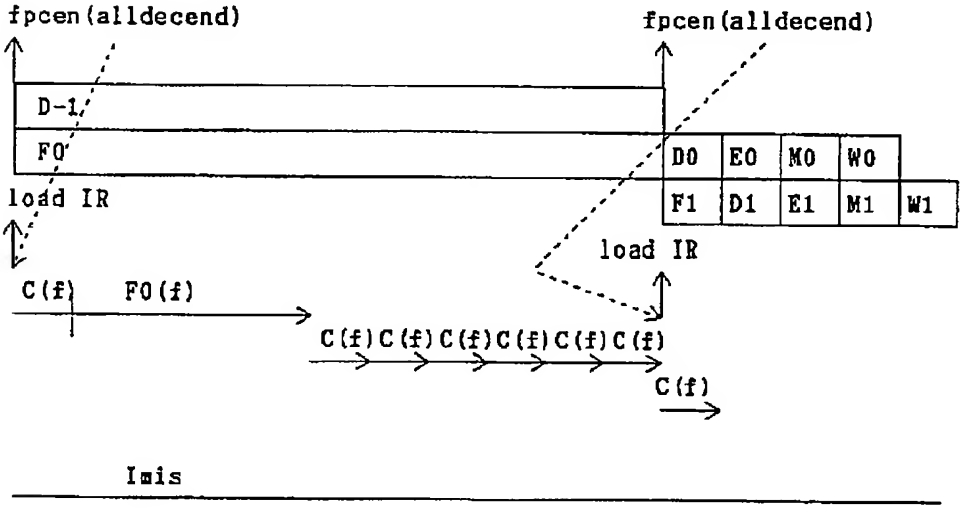
【図４】


メインメモリ
I/O装置
命令キャッシュメモリ
命令発行ユニット
マルチポートファイル
2ポートデータキャッシュメモリ
MA0
MA1
EC1
EC2
ALU0
ALU1
FADD
FMUL
アボートアドレスレジスタ
トラップアドレスレジスタ
トラップ原因レジスタ
トラップ制御ユニット
ストール制御ユニット
トラップ信号

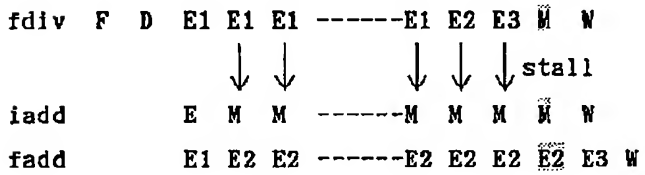
【図６】


In
In+1
In+2
In+3
Fn
Dn
E (n)
M (n)
W (n)
E (n+1)
M (n+1)
W (n+1)
E (n+2)
M (n+2)
M (n+2)
W (n+2)
E (n+3)
M (n+3)
M (n+3)
W (n+3)
F (n+4)
D (n+4)
E (n+4)
E (n+4)
M (n+4)
W (n+4)
E (n+5)
E (n+5)
M (n+5)
W (n+5)
E (n+6)
E (n+6)
M (n+6)
W (n+6)
E (n+7)
E (n+7)
M (n+7)
W (n+7)
F (n+8)
D (n+8)
E (n+8)
M (n+8)
W (n+8)
E (n+9)
M (n+9)
W (n+9)
E (n+10)
M (n+10)
W (n+10)
E (n+11)
M (n+11)
W (n+11)
C
C+1
C+2
C+3
C+4
C+5
C+6

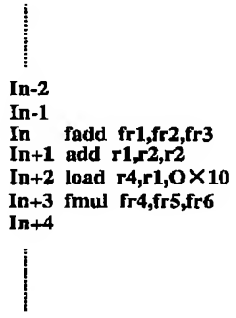
【図９】


fpcen(alldecend)
fpcen(alldecend)
D-1
F0
load IR
D0
E0
M0
W0
F1
D1
E1
M1
W1
load IR
C(f)
F0(f)
C(f)C(f)C(f)C(f)C(f)C(f)
C(f)
Imis

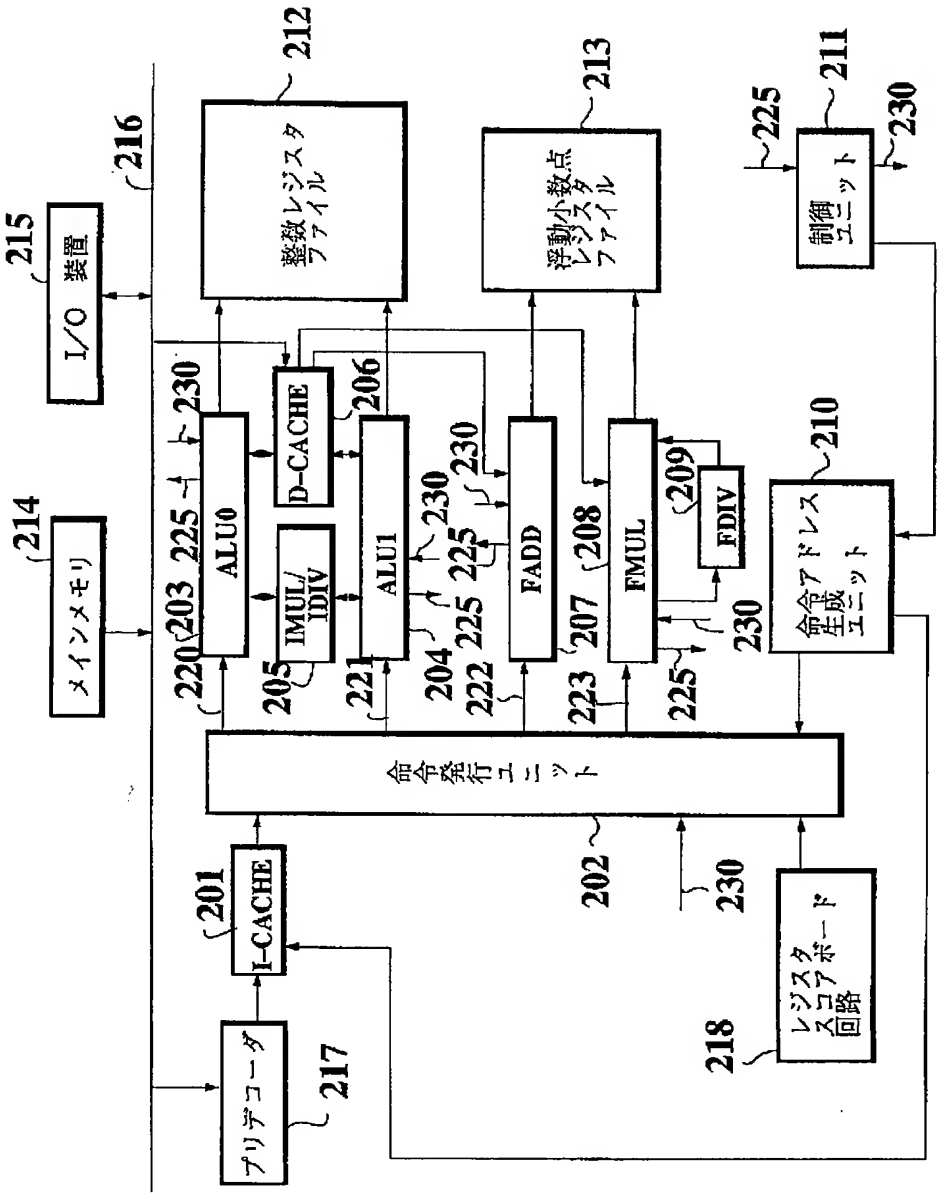
【図１４】


fdiv F D E1 E1 E1 ------E1 E2 E3 M W
stall
iadd E M M ------M M M M W
fadd E1 E2 E2 ------E2 E2 E2 E2 E3 W

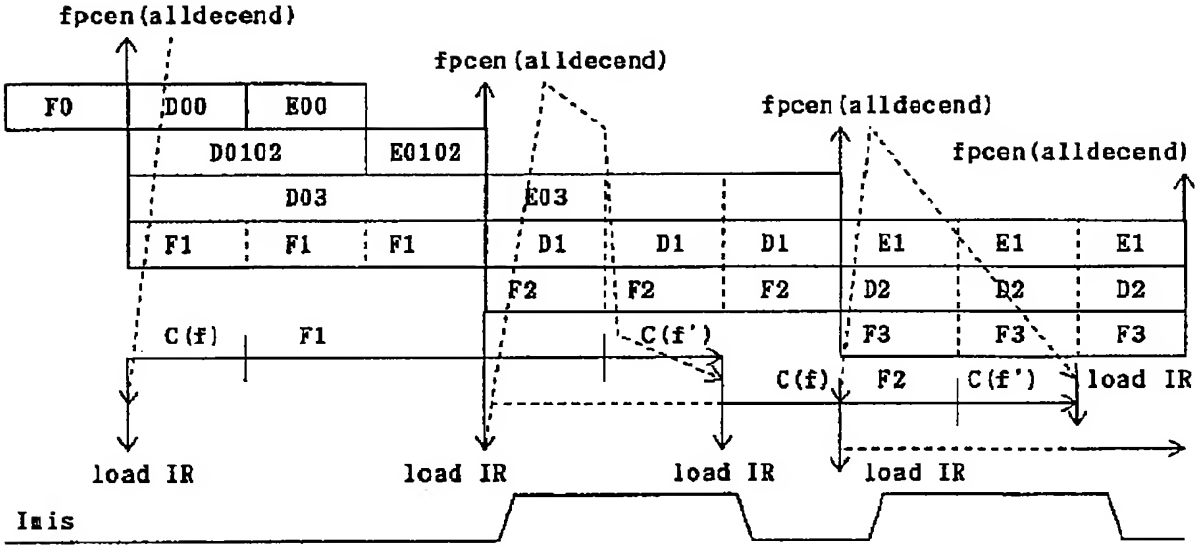
【図２１】


In-2
In-1
In fadd fr1,fr2,fr3
In+1 add r1,r2,r2
In+2 load r4,r1,0×10
In+3 fmul fr4,fr5,fr6
In+4

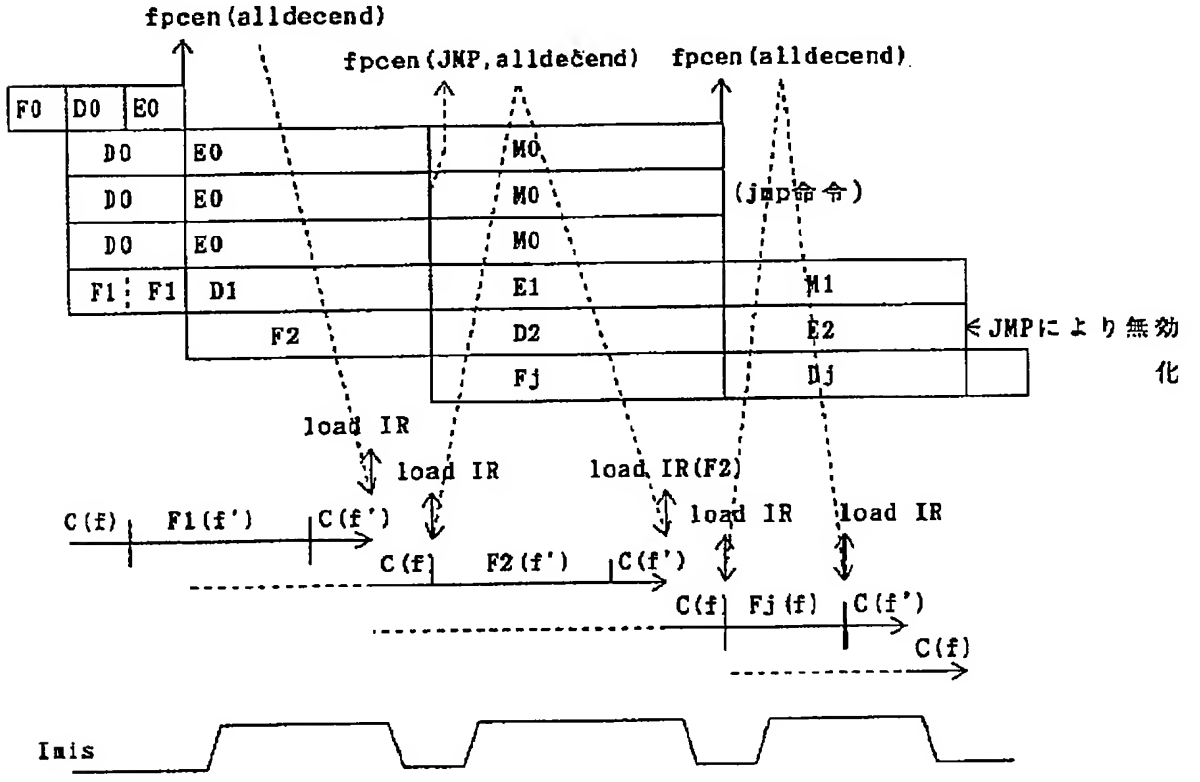
【図７】


整数レジスタファイル
212
浮動小数点レジスタファイル
213
制御ユニット
211
225
230
I/O 装置
215
216
メインメモリ
214
D-CACHE
206
ALU0
203
225
230
IMUL/IDIV
205
ALU1
221
230
FADD
204
225
222
FMUL
207
208
FDIV
209
225
223
230
命令アドレス生成ユニット
210
命令発行ユニット
202
230
I-CACHE
201
プリデコーダ
217
レジスタスコアボード回路
218

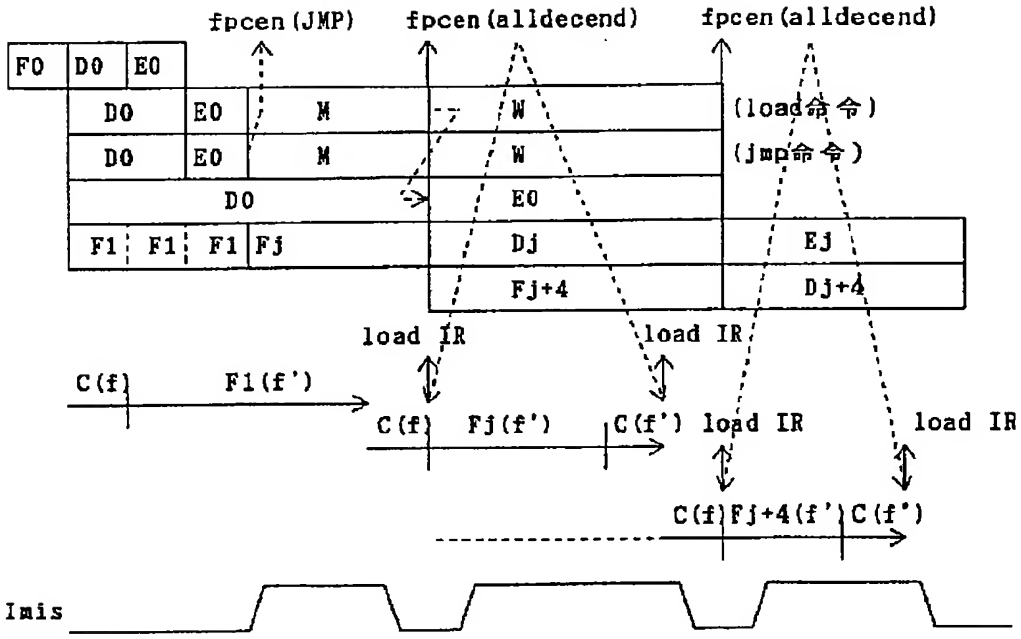
【図１０】


fpcen(alldecend)
fpcen(alldecend)
fpcen(alldecend)
fpcen(alldecend)
F0
D00
E00
D0102
E0102
D03
E03
F1
F1
F1
D1
D1
D1
E1
E1
E1
F2
F2
F2
D2
D2
D2
F3
F3
F3
C(f)
F1
C(f')
C(f)
F2
C(f')
load IR
load IR
load IR
load IR
load IR
Imis

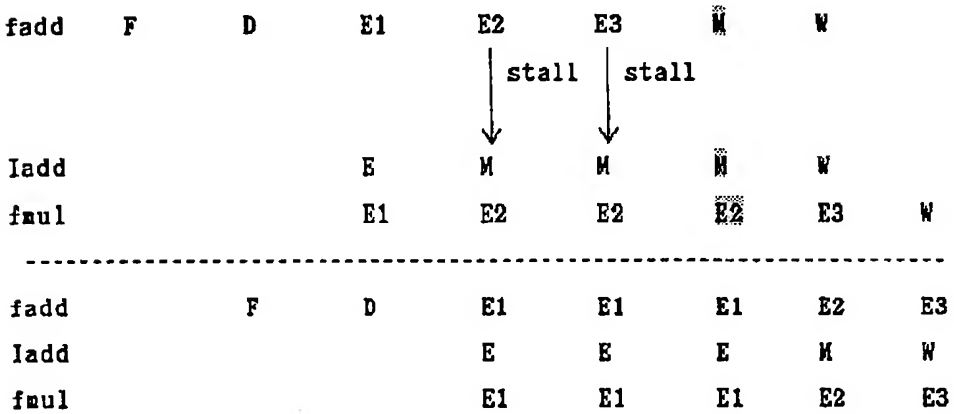
【図１１】


fpcen(alldecend)
fpcen(JMP,alldecend)
fpcen(alldecend)
F0
D0
E0
D0
E0
M0
D0
E0
M0
(jmp命令)
D0
E0
M0
F1
F1
D1
E1
M1
F2
D2
E2
≪JMPにより無効化
Fj
Dj
load IR
load IR
load IR(F2)
load IR
load IR
C(f)
F1(f')
C(f')
C(f)
F2(f')
C(f')
C(f)
Fj(f)
C(f')
C(f)
Imis

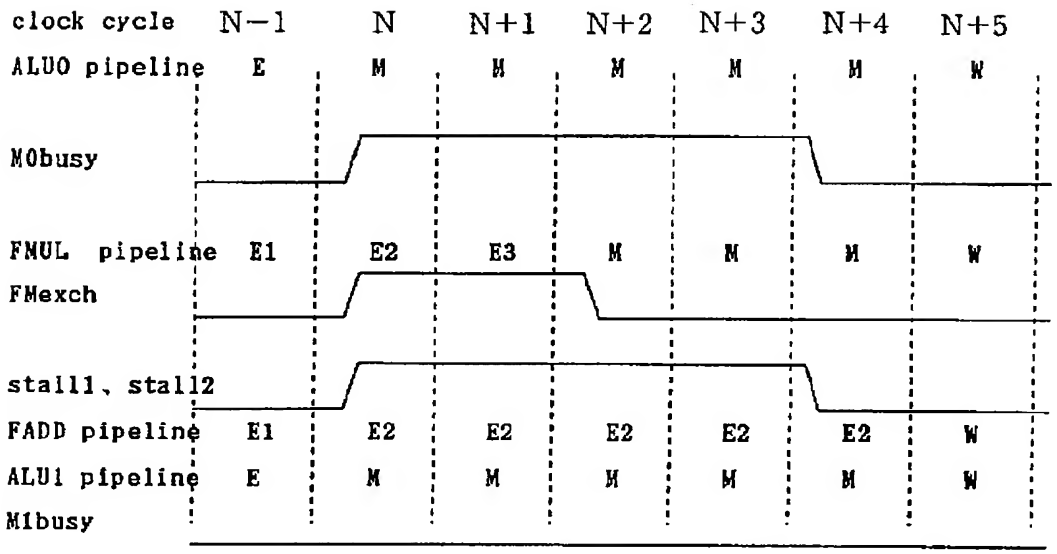
【図１２】


fpcen(JMP)
fpcen(alldecend)
fpcen(alldecend)
F0
D0
E0
D0
E0
M
W
(load命令)
D0
E0
M
W
(jmp命令)
D0
E0
F1
F1
F1
Fj
Dj
Ej
Fj+4
Dj+4
load IR
load IR
C(f)
F1(f')
C(f)
Fj(f')
C(f')
load IR
load IR
C(f)Fj+4(f')C(f')
Imis

【図１３】

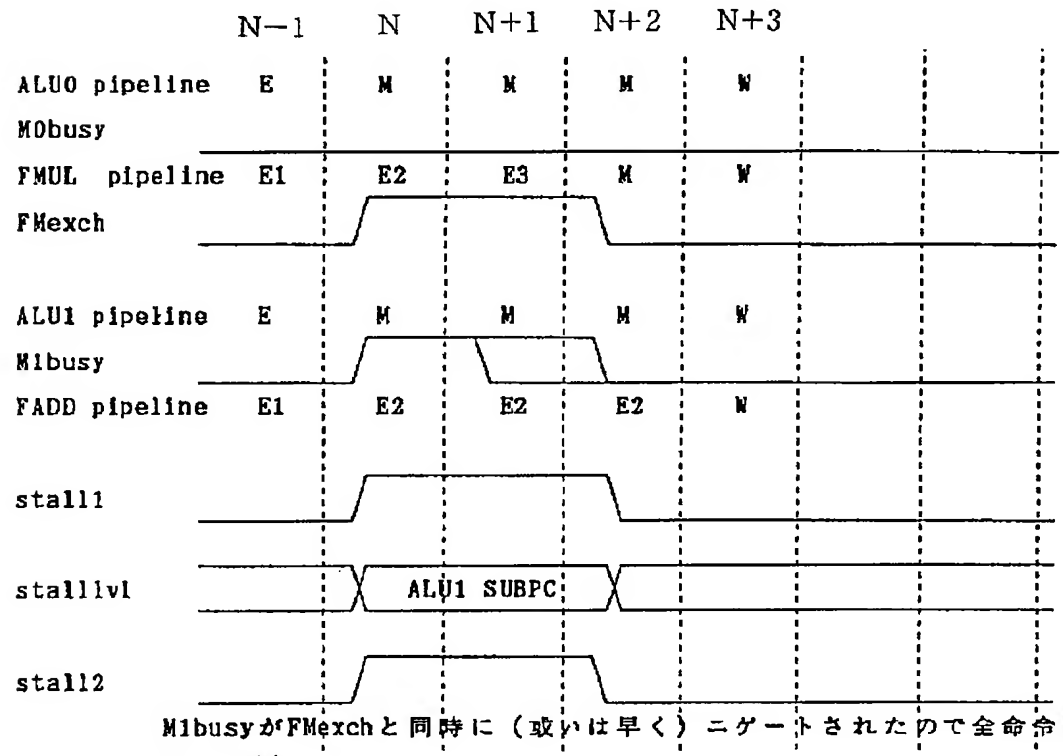

fadd F D E1 E2 E3 M W
stall
stall
Iadd E M M M W
fmul E1 E2 E2 E2 E3 W
fadd F D E1 E1 E1 E2 E3
Iadd E E E M W
fmul E1 E1 E1 E2 E3

【図１５】



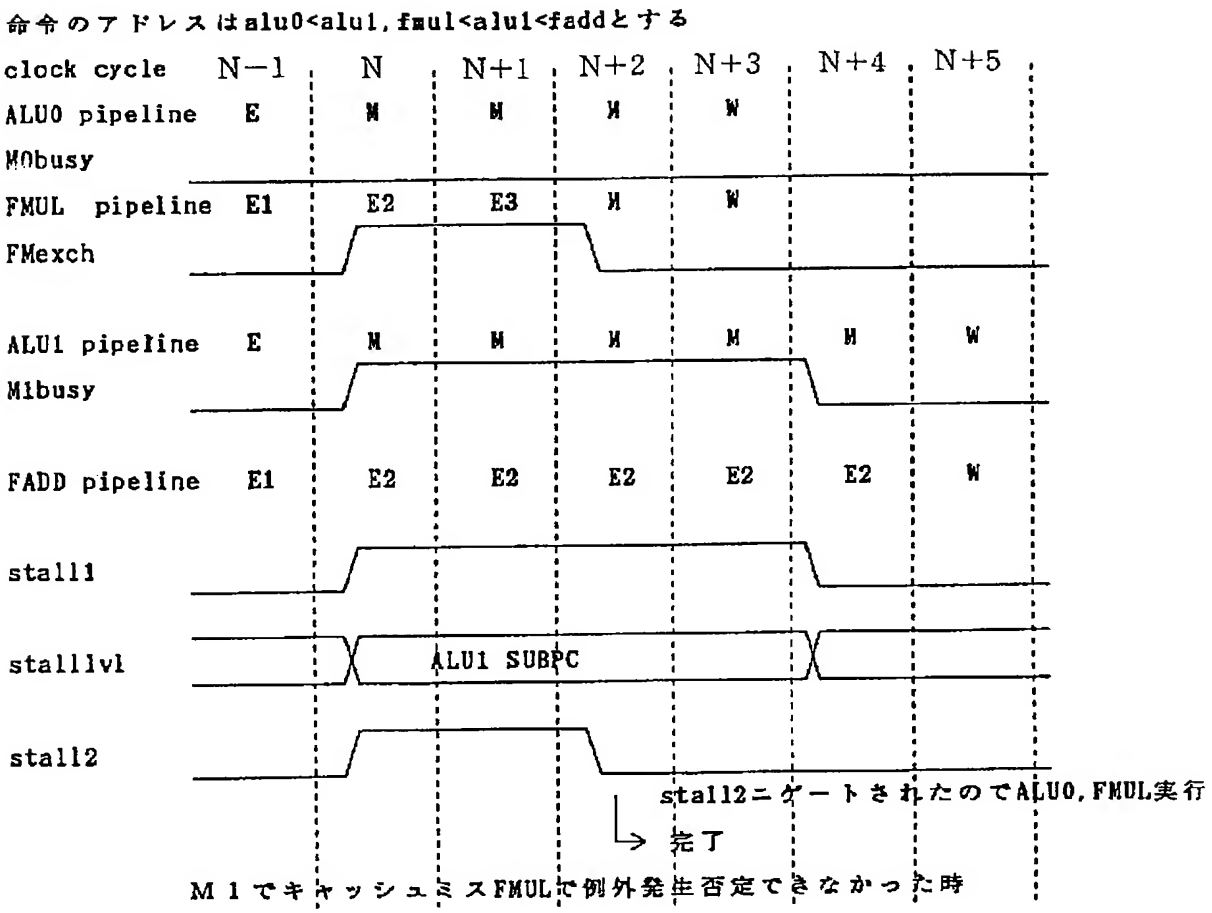
Ｍ０でキャッシュミスFMULで例外発生否定できなかった時

【図１７】



M1busyがFMexchと同時に（或いは早く）ニゲートされたので全命令同時実行完了した例

【図１６】


命令のアドレスはalu0<alu1,fmul<alu1<faddとする
clock cycle
N−1
N
N+1
N+2
N+3
N+4
N+5
ALU0 pipeline
E
M
M
M
W
M0busy
FMUL pipeline
E1
E2
E3
M
W
FMexch
ALU1 pipeline
E
M
M
M
M
M
W
M1busy
FADD pipeline
E1
E2
E2
E2
E2
E2
W
stall1
stall1v1
ALU1 SUBPC
stall2
stall2ニゲートされたのでALU0,FMUL実行
完了
M１でキャッシュミスFMULで例外発生否定できなかった時

【図１８】

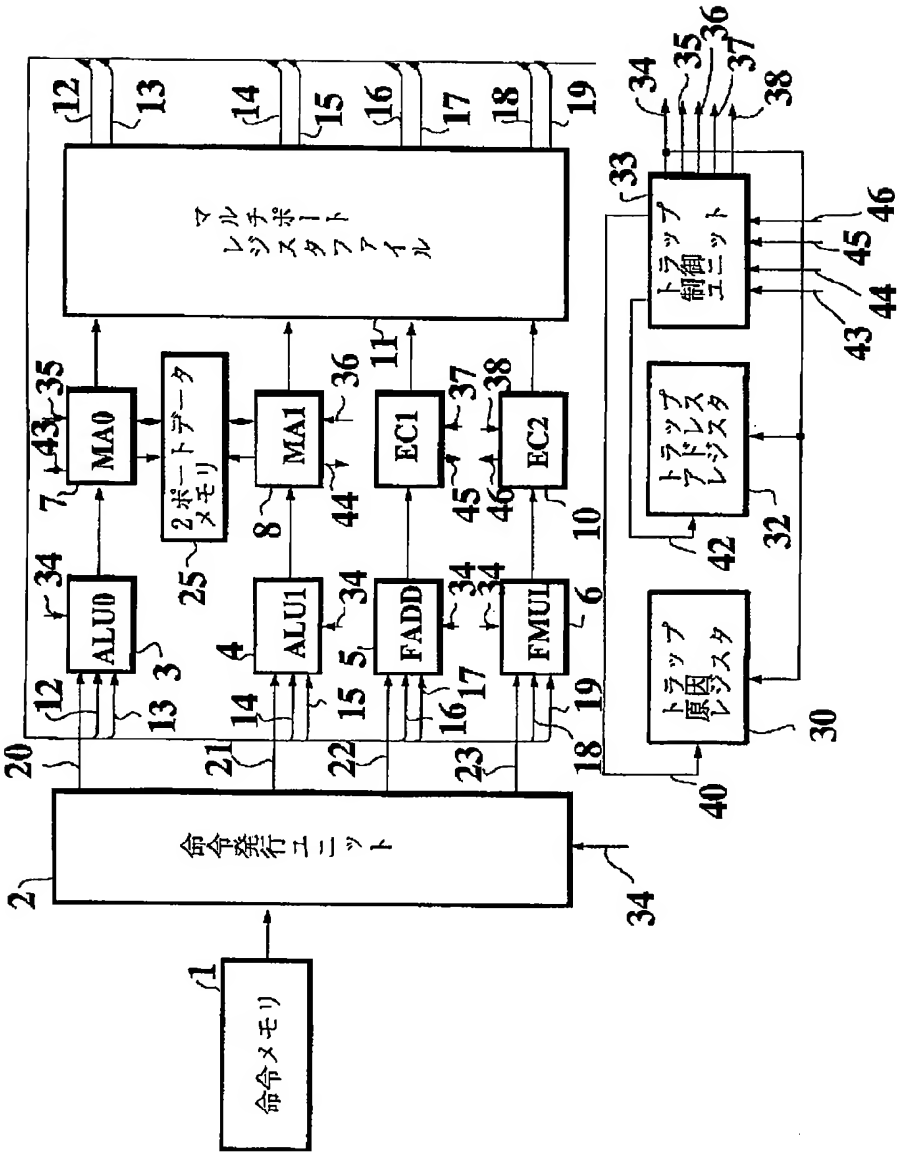
stall1		0	0	0	0	1	1	1	1	備考
stall2		0	0	1	1	0	0	1	1	
FRbusy		0	1	0	1	0	1	0	1	
F stage		○	有り得ない			×	有り得ない	×	×	
D stage		○				×		×	×	
ALU	E stage	○				×		×	×	
	M stage0	○				○		×	×	実行完了stage
	M stage1	○				×		×	×	実行完了stage
	W stage	○				○		○	○	
MUL/DIV	E1$_1$ stage	○				×		×	×	
	E1$_2$ stage	○				△		×	×	実行完了stage
	E1$_L$ stage	○				○		○	○	
	E1$_v$ stage	○				○		○	○	
FADD (FMUL) FAexch= NEGATE	E1	○				×		×	×	
	E2	○				△		×	×	実行完了stage
	E3	○				○		○	○	
	W stage	○				○		○	○	
FADD (FMUL) FAexch= ASSERT	E1 stage	○				×		×	×	
	E2 stage	○				○		○	×	← FADDは動作も可
	E3 stage	○				○		○	○	
	M stage	○				△		×	×	実行完了stage
	W stage	○				○		○	○	
FDIV FAexch= NEGATE	E1$_1$ stage	○				×		×	×	
	E1$_2$ stage	○				△		×	×	実行完了stage
	E1$_L$ stage	○				○		○	○	
	E2 stage	○				○		○	○	
	E3 stage	○				○		○	○	
	W stage	○				○		○	○	
FDIV FAexch= ASSERT	E1$_1$ stage	○				×		×	×	
	E1$_2$ stage	○				○		○	○	
	E1$_L$ stage	○				○		○	○	
	E2 stage	○				○		○	○	
	E3 stage	○				○		○	○	
	M stage	○				△		×	×	実行完了stage
	W stage	○				○		○	○	
BU	E stage	○				×		×	×	
	M stage	○				△		×	×	
	W stage	○				○		○	○	

○：ストールせず

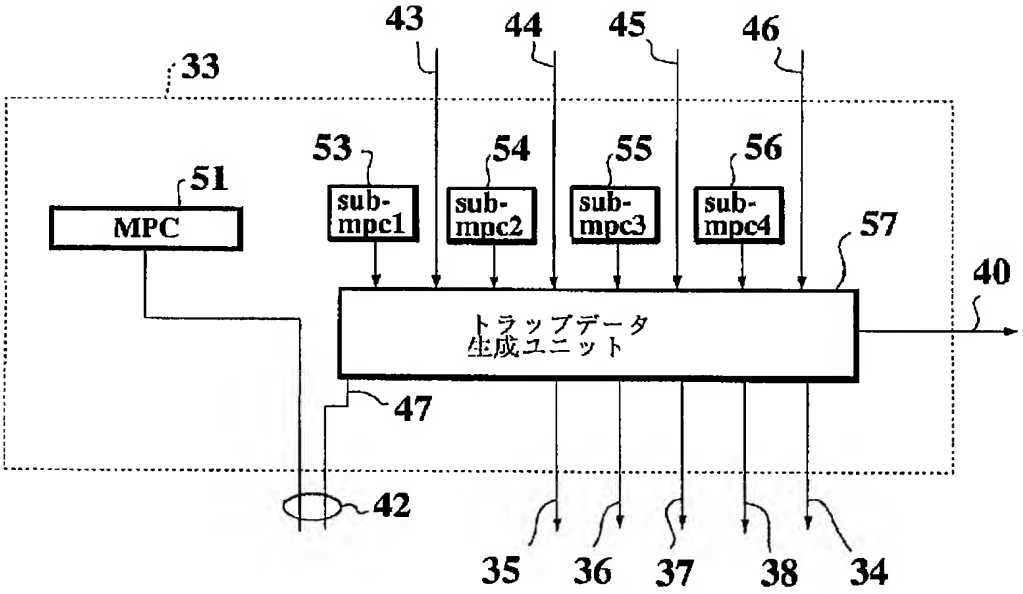
×：ストールする

△：ｓｔｌｌｖｌ＞ｓｕｂｐｃならストールせず、そうでなければストールする

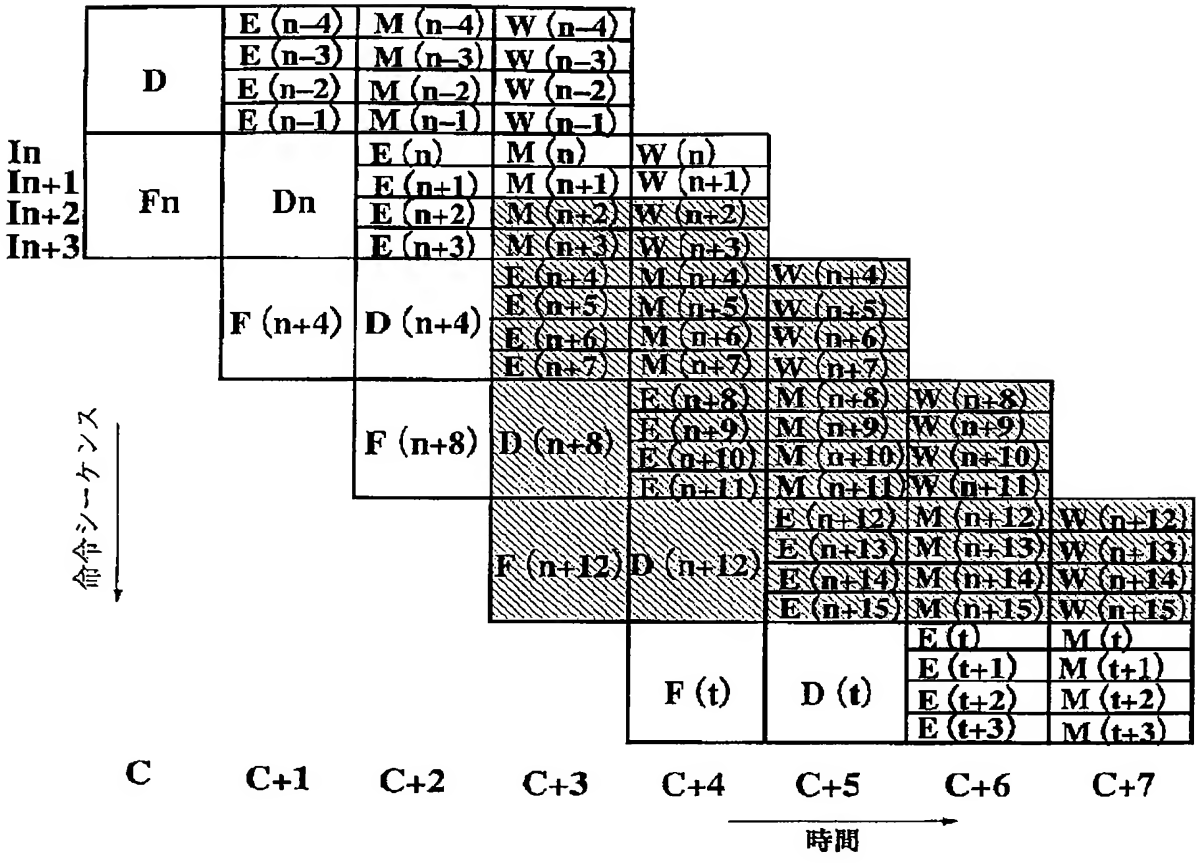
【図１９】


命令メモリ
命令発行ユニット
ALU0
ALU1
FADD
FMUl
MA0
MA1
EC1
EC2
2ポートデータメモリ
マルチレジスタファイル
トラップ制御ユニット
トラップアドレスレジスタ
トラップ原因レジスタ

【図２０】


43
44
45
46
33
53
54
55
56
51
MPC
sub-mpc1
sub-mpc2
sub-mpc3
sub-mpc4
57
40
トラップデータ生成ユニット
47
42
35
36
37
38
34

【図２２】


D
Fn
Dn
F (n+4)
D (n+4)
F (n+8)
D (n+8)
F (n+12)
D (n+12)
F (t)
D (t)
E (n–4)
M (n–4)
W (n–4)
E (n–3)
M (n–3)
W (n–3)
E (n–2)
M (n–2)
W (n–2)
E (n–1)
M (n–1)
W (n–1)
In
In+1
In+2
In+3
E (n)
M (n)
W (n)
E (n+1)
M (n+1)
W (n+1)
E (n+2)
M (n+2)
W (n+2)
E (n+3)
M (n+3)
W (n+3)
E (n+4)
M (n+4)
W (n+4)
E (n+5)
M (n+5)
W (n+5)
E (n+6)
M (n+6)
W (n+6)
E (n+7)
M (n+7)
W (n+7)
E (n+8)
M (n+8)
W (n+8)
E (n+9)
M (n+9)
W (n+9)
E (n+10)
M (n+10)
W (n+10)
E (n+11)
M (n+11)
W (n+11)
E (n+12)
M (n+12)
W (n+12)
E (n+13)
M (n+13)
W (n+13)
E (n+14)
M (n+14)
W (n+14)
E (n+15)
M (n+15)
W (n+15)
E (t)
M (t)
E (t+1)
M (t+1)
E (t+2)
M (t+2)
E (t+3)
M (t+3)
命令シーケンス
C
C+1
C+2
C+3
C+4
C+5
C+6
C+7
時間

フロントページの続き

(72)発明者　武田　譲治
神奈川県川崎市幸区小向東芝町１　株式会
社東芝総合研究所内

【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第６部門第３区分
【発行日】平成１１年（１９９９）１１月３０日

【公開番号】特開平５－１８１６７６
【公開日】平成５年（１９９３）７月２３日
【年通号数】公開特許公報５－１８１７
【出願番号】特願平４－８２４９０
【国際特許分類第６版】
G06F 9/38 380
310
370
11/00 310
15/16 390
【ＦＩ】
G06F 9/38 380 B
310 X
370 X
11/00 310 G
15/16 390 Z

【手続補正書】
【提出日】平成１１年４月２日
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】特許請求の範囲
【補正方法】変更
【補正内容】
【特許請求の範囲】
【請求項１】　複数シーケンスの命令を同時実行するＮ個（Ｎは正整数）の命令実行手段と、
前記Ｎ個の命令実行手段により同時実行される命令を供給する命令供給手段と、
Ｍ個（Ｎ≧Ｍ、Ｍは正整数）の命令が前記命令供給手段から前記Ｎ個の命令実行手段に同時に供給され、一クロックサイクル中に前記Ｍ個の命令の中の少なくとも一つの命令実行において処理例外が発生したとき、該一クロックサイクル中に前記Ｎ個の命令実行手段の全てにアボート信号を送ることにより前記Ｎ個の命令実行手段に供給された前記Ｍ個の命令の同時処理を全て該一クロックサイクル中にアボートするように前記Ｎ個の命令実行手段を制御するトラップ制御手段と、
を備えたことを特徴とする命令レベル並列処理型プロセッサシステム。
【請求項２】　前記トラップ制御手段により処理がアボートされた前記Ｍ個の命令の中で最も小さいアドレスを有する命令のアドレスを格納するアボートアドレス格納手段と、
前記Ｍ個の命令の中で前記処理例外を発生させた命令のアドレスを格納するトラップアドレス格納手段と、
を更に備えたことを特徴とする請求項１記載の命令レベル並列処理型プロセッサシステム。
【請求項３】　前記Ｎ個の命令実行手段は、Ｋ個（Ｍ≧Ｋ、Ｋは整数）の同等の機能を有する順序付けされた演算手段を有し、前記命令供給手段はＪ個（Ｋ≧Ｊ＞１、Ｊは整数）の順序付けされた命令を、該Ｊ個の命令の中でより前の順序にある命令が該Ｋ個の演算手段の中でより前の順序にある演算手段に供給されるように、該Ｋ個の演算手段に供給することを特徴とする請求項１記載の命令レベル並列処理型プロセッサシステム。
【請求項４】　前記Ｊ個の命令中のＩ番目の命令の実行において処理例外が発生する可能性を否定しきれないときには、前記Ｊ個の命令の中で該Ｉ番目の命令よりも後の順序にある命令の処理をストールするように、前記Ｊ個の命令の一部の処理をストールするストール手段を、更に備えたことを特徴とする請求項３記載の命令レベル並列処理型プロセッサシステム。
【請求項５】　前記Ｍ個の命令の実行において処理例外が発生する可能性を否定しきれないときには、前記Ｍ個の命令の処理をストールし、前記Ｍ個の命令の実行において処理例外が実際に発生したときには、前記Ｍ個の命令の処理をアボートし、前記Ｍ個の命令の実行において処理例外が実際に発生しなかったときには、前記Ｍ個の命令の処理を再開するストール制御手段を、更に備えたことを特徴とする請求項１記載の命令レベル並列処理型プロセッサシステム。
【請求項６】　前記処理例外に対処する手段と、
前記処理例外に対処した後に、前記トラップ制御手段に

よりアボートされた前記M個の命令を同時に再スタートする手段と、
を更に備えたことを特徴とする請求項１記載の命令レベル並列処理型プロセッサシステム。
【請求項７】　命令レベル並列処理型プロセッサシステムの制御方法であって、
該システムのＮ個（Ｎは正整数）の命令実行手段により同時実行される複数シーケンスのM個（Ｎ≧M、Mは正整数）の命令を供給するステップと、
前記M個の命令が前記Ｎ個の命令実行手段に同時に供給され、一クロックサイクル中に前記M個の命令の中の少なくとも一つの命令実行において処理例外が発生したとき、該一クロックサイクル中に前記Ｎ個の命令実行手段の全てにアボート信号を送ることにより前記Ｎ個の命令実行手段に同時に供給された前記M個の命令の処理を全て該一クロックサイクル中にアボートするように前記Ｎ個の命令実行手段を制御するステップと、
を備えたことを特徴とする命令レベル並列処理型プロセッサシステムの制御方法。